National Library of Canada
Bibliothèque nationale du Canada

Canadian Theses Service
Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

NL-339 (r. 88/04) c

Canada

University of Alberta

MULTIPLE SOLUTIONS OF PLANE ELASTICAE

by

Victor H.Y. Tam

A Thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Master of Science**.

Department of Mechanical Engineering

Edmonton, Alberta

Fall 1991

National Library of Canada
Bibliothèque nationale du Canada

Canadian Theses Service
Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-70067-X

Canada

University of Alberta
Release Form

Name of Author: Victor Hoi Yiu Tam

Title of Thesis: MULTIPLE SOLUTIONS OF PLANE ELASTICAE

Degree: Master of Science

Year this degree granted: Fall 1991

Permission is hereby granted to **The University of Alberta Library** to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly, or scientific research purposes only.

The author reserves all other publication and other rights in association with the copyright in the thesis, and except as hereinbefore provided neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatever without the author's written consent.

Victor H.Y. Tam

T3-1022, 216 Crown Road
Edmonton, Alberta
Canada T6J 2E3

Date: October 9, 1991.

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **MULTIPLE SOLUTIONS OF PLANE ELASTICAE** submitted by **Victor H.Y. Tam** in partial fulfillment of the requirements for the degree of **Master of Science**.

Gary Faulkner

M.G. Faulkner (Supervisor)

A.W. Lipsett

A.W. Lipsett (Supervisor)

David Steigmann

D.J. Steigmann

T. Brudey

T.M. Brudey

Date: Oct. 1, 1991

To My Parents

ABSTRACT

The large deflections of inextensible elastic rods have always been of interest and because of the severe nonlinearity of the problem, various numerical and analytical formulations have emerged over the years.

In this study a general numerical method, the segmental shooting technique, is employed to investigate the non-uniqueness of these nonlinear elastica problems. Specific problems, including cantilevers under free end concentrated load, distributed load and normal load as well as deep arches loaded by concentrated loads and with various boundary conditions are considered. Where other solutions are available, the segmental shooting technique is shown to be in close agreement.

The determination of the number of possible deformed configurations for a given load is also investigated for cantilevers and arches. It is found in general, that the number of multiple solutions increases as the applied load is increased. The stability of these deformed configurations is examined by using the segmental approach based on the criterion of minimum potential energy. This is possible as the segmental technique employed is well suited to generate a series of kinematically admissible displacement fields.

ACKNOWLEDGEMENT

The author would like to take this opportunity to express his deepest appreciation to the people who contributed their valuable time, advice and resources towards this thesis. These distinct individuals in random order include:

Dr. M. G. Faulkner for his valuable advice, guidance and encouragement.

Dr. A. W. Lipsett for his utmost patience, innovation as well as his dedication.

Dr. D. J. Steigmann for a continual fresh perspective into the subject.

Mr. S. P. McEvoy for being my mentor and best friend while much has to be learned from his confidence, enthusiasm and positivism.

Mr. R. T. Mah for his perpetual optimism and charm.

Ms. Carmen Chan for her tolerance, patience and admirable personality bearing with me through the tough times.

Ms. Georgeana Wong for being one of my best friends as well as my good teacher.

Lastly, thanks you to all other fine individuals including Dr. J. Wolffaardt, Kosay El-Rayes, Don Raboud, Jeffery Kemps, the girls on the fourth floor (Helen, Betty, Gail, Doris and Betty-Ann), the machinists especially Al Muir and Max Schubert, the technicians, the Department of Mechanical Engieering and anybody else that I might have missed. It has been both my honour and pleasure working with them.

Table of Contents

CHAPTER ONE

INTRODUCTION 1

1.1 Background 1

1.2 Thesis Outline 4

CHAPTER TWO

THEORETICAL FORMULATION 6

2.1 Equilibrium Equations 7

2.2 Segmental Technique 18

2.3 Shooting Method 31

CHAPTER THREE

VERIFICATION 36

3.1 Point Load 38

3.2 Uniform Distributed Vertical Load w_y 45

3.3 Normal Load w_n 46

3.4 Pin End Deep Arch 53

CHAPTER FOUR

RESULTS AND DISCUSSIONS 56

4.1 Minimum Potential Energy Criterion for Stability 57

4.2 Energy Considerations of Cantilevers 59

4.2.1 Cantilever Under Free End Concentrated Load 59

4.2.2 Cantilever Under Distributed Load 64

4.3 Arches 66

4.3.1 Zero Load Solution 67

4.3.2 Arch with Concentrated End Load 71

4.3.3 Zero Solution(s) Revisited 85

4.3.4 Arch with Concentrated Crown Load 90

4.3.5 Pin-pin Supported Arch with Concentrated Crown Load 104

CHAPTER FIVE

CONCLUDING REMARKS 107

5.1 Summary 107

5.2 Recommendations 110

REFERENCES 112

APPENDIX A

ANALYTICAL SOLUTIONS OF THE CANTILEVER UNDER FREE END CONCENTRATED LOAD 115

APPENDIX B

ANALYTICAL SOLUTIONS OF THE CANTILEVER UNDER UNIFORM NORMAL LOAD 122

APPENDIX C

SURFACE PLOTS ACCURACY 127

APPENDIX D

SEGMENTAL SHOOTING TECHNIQUE SOURCE LISTING 130

LIST OF TABLES

Table 2.1	Nodal Connectivity between Successive Segments	22
Table 3.1	Analytical Results of Cantilever Subject to a Free End Point Load P (Geometry of Free End)	41
Table 3.2	Numerical Results of Cantilever Subject to a Free End Point Load P (Geometry of Free End)	42
Table 3.3	Analytical and Numerical Results of a Cantilever Subject to Uniform Normal Load (Geometry of Free End)	49
Table 3.4	Analytical and Numerical Results of a Cantilever Subject to Uniform Normal Load q_n = 10 & 15 (Geometry of Free End)	51
Table 4.1	Multiple Equilibrium Solutions of the Arch for q = 1.0, 2.0 3.0 and 4.0	79
Table 4.2	Multiple Equilibrium Solutions of the Arch for q = 4.0, 5.0 and 6.0	80

LIST OF FIGURES

Figure 2.1	Free and Deformed Shape of the Rod	8
Figure 2.2	Infinitesimal Element Loaded by Horizontal and Vertical Distributed Load	9
Figure 2.3	Infinitesimal Element Loaded by Normal and Tangential Distributed Load	12
Figure 2.4	Application of the Segmental Approach	19
Figure 2.5	Local Coordinate System of an Element	20
Figure 2.6	Numerous Linear Segments Assembled to Form a Nonlinear Problem	21
Figure 2.7	Geometric Compatibility of Successive Segments	23
Figure 2.8	Force Compatibility of Successive Segments	24
Figure 2.9	Flow Diagram of Numerical Shooting Technique	35
Figure 3.1	Cantilever Under Various Loading Conditions	37
Figure 3.2	Free Node Moment Λ_2 vs Fixed Node Moment Λ_1 of the Cantilever Under Concentrated End Load	44
Figure 3.3	Multiple Equilibrium Solutions of a Cantilever Under Free End Load	44
Figure 3.4	Free End Moment Λ_2 vs Fixed End Moment Λ_1 of the Cantilever Under Vertical Distributed Load	45
Figure 3.5	Multiple Equilibrium Solutions of Cantilever Under Vertical Distributed Load	46
Figure 3.6	Mitchell's Notation for Cantilever Under Normal Load	48
Figure 3.7	Fixed Node Moment Λ_1 vs Fixed Node Slope γ_1 of the Cantilever Under Normal Load	50
Figure 3.8	Analytical and Numerical Results of the Cantilever Under Normal Load	51
Figure 3.9	Analytical and Numerical Results of the Cantilever Under Normal Load	52

Figure 3.10	Boundary Conditions of the Semi-Circular Arch	53
Figure 3.11	Pin End Semi-Circular Arch Subject to Horizontal Mid-Span Point Load	54
Figure 3.12	Free and Deformed Shape of the Arch Under Horizontal Crown Load	55
Figure 4.1	Multiple Solutions of a Cantilever Under Free End Point Load	60
Figure 4.2	Free End Moment Λ_2 and Energy Π vs Fixed End Moment Λ_1 For Cantilever Loaded at Free End	61
Figure 4.3	Potential Energy For Concentrated Loads	63
Figure 4.4	Multiple Solutions of the Cantilever Under Vertical Distributed Load	65
Figure 4.5	Free End Moment Λ_2 and Energy Π vs Fixed End Moment Λ_1 of Cantilever Under Distributed Load	65
Figure 4.6	Potential Energy For Uniformly Distributed Loads	66
Figure 4.7	Semi-Circular Arch with Pin-Roller Supports	67
Figure 4.8	Possible Zero Load Solution(s) of Pin-Roller Supported Arch	68
Figure 4.9	Rigid Body Rotation of Arch with No Change in Strain Energy	69
Figure 4.10	Positive Prescribed Moments at Both Ends Satisfying Boundary Condition $Y_2 = 0$	69
Figure 4.11	Potential Energy Π of an Unloaded Arch	70
Figure 4.12	Semi-Circular Arch Subject to a Horizontal Point Load P Applied at The Roller Support	72
Figure 4.13	Combined End Moment Λ_2, $Y_2/2R$ vs Start Slope γ_1 of the Pin-Roller Support Arch	73
Figure 4.14	Deformed and Undeformed Shape of the Arch for $q = 1.0$	73
Figure 4.15	Positive and Negative Prescribed Moments at Both Ends Satisfying Boundary Condition $Y_2 = 0$	74
Figure 4.16	Excessive Prescribed Positive Moments Resulting in a "*Spiral*"	75

Figure 4.17 Potential Energy Π of Arch for q = 0.0 and 1.0 77

Figure 4.18 $Y_2/2R$ vs Start Slope γ_1 of the Arch for q = 2.0, 3.0 & 4.0. 77

Figure 4.19 Λ_2 vs Start Slope γ_1 of the Arch for q = 2.0, 3.0 & 4.0. 77

Figure 4.20 Deformed and Undeformed Shapes of the Arch for q = 2.0, 3.0 and 4.0. 78

Figure 4.21 Potential Energy Π vs Start Slope γ_1 of the Pin-Roller Support Arch 79

Figure 4.22 Combined End Moment Λ_1, $Y_2/2R$ vs Start Slope γ_1 of the Pin-Roller Support Arch 80

Figure 4.23 Deformed and Undeformed Shapes of Arch for q = 5.0 81

Figure 4.24 Potential Energy Π of Arch for q = 5.0 82

Figure 4.25 Combined End Moment Λ_2, $Y_2/2R$ vs Start Slope γ_1 Plot of the Pin-Roller Support Arch for q = 6.0 84

Figure 4.26 Deformed and Undeformed Shapes of Arch for q = 6.0 84

Figure 4.27 Potential Energy Π of Arch for q = 6.0 85

Figure 4.28 Free Body of the Semi-Circular Arch after Deformation 86

Figure 4.29 Possible Zero Load Solutions for the Pin-Roller Arch 88

Figure 4.30 Free Body Diagram of the Arch Under a Horizontal End Load 89

Figure 4.31 Semi-Circular Arch Under a Concentrated Crown Load 90

Figure 4.32a Contour Plot For q = 1.0 ($\pi/30 < \gamma_1 < 29\pi/30$) 92

Figure 4.32b Contour Plot for q = 1.0 ($-29\pi/30 < \gamma_1 < -\pi/30$) 93

Figure 4.33 Multiple Solutions of the Arch With Crown Load q = 1.0 94

Figure 4.34a Contour Plot for q = 1.2 ($\pi/30 < \gamma_1 < 29\pi/30$) 97

Figure 4.34b Contour Plot for q = 1.2 ($-29\pi/30 < \gamma_1 < -\pi/30$) 97

Figure 4.35a Contour Plot for q = 1.4 ($\pi/30 < \gamma_1 < 29\pi/30$) 98

Figure 4.35b Contour Plot for q = 1.4 ($-29\pi/30 < \gamma_1 < -\pi/30$) 98

Figure 4.36a Contour Plot for q = 1.6 ($\pi/30 < \gamma_1 < 29\pi/30$) 99

Figure 4.36b Contour Plot for q = 1.6 ($-29\pi/30 < \gamma_1 < -\pi/30$) 99

Figure 4.37 Multiple Solutions of the Arch with Crown Load q = 1.2, 1.4 and 1.6 100

Figure 4.38a Contour Plot for q = 1.8 ($\pi/30 < \gamma_1 < 29\pi/30$) 101

Figure 4.38b Contour Plot for q = 1.8 ($-29\pi/30 < \gamma_1 < -\pi/30$) 101

Figure 4.39 Multiple Solutions of the Arch with Crown Load q = 1.8 102

Figure 4.40a Contour Plot for q = 2.0 ($\pi/30 < \gamma_1 < 29\pi/30$) 102

Figure 4.40b Contour Plot for q = 2.0 ($-29\pi/30 < \gamma_1 < -\pi/30$) 103

Figure 4.41 Multiple Solutions of the Arch with Crown Load q = 2.0 103

Figure 4.42 Semi-Circular Arch Pinned at Both Ends 104

Figure 4.43 Multiple Solutions of the Pin-Pin Arch with Crown Load (α = 90°) 105

Figure 4.44 Multiple Solutions of the Pin-Pin Arch with Crown Load (α = 135°) 106

Figure C.1 Contour of a Hemisphere for Z = 0.5 and 1.0 using 100 X 100 Normal Search Grid 128

Figure C.2 Contour of a Hemisphere for Z = 0.5 and 1.0 using 200 X 200 Octant Search Grid 129

NOMENCLATURE

$\underline{A}$	A vector containing the initial knowns and unknowns at the initial node
$\underline{B}$	A vector containing the known boundary value(s) at the end node
$\underline{b}$	body force
$\underline{C}$	A vector containing the computed boundary value(s) at the end node
C_1, C_2	Constants of integration
$\underline{E}$	Error function vector given by the difference of **B** and **C**
EI	The flexural rigidity of the rod
$\underline{F}$	Applied force
$\underline{J}$	Jacobian matrix
L	Length of the rod; load potential of the rod
M	Bending moment
P	Applied concentrated load
R	Radius of curvature
T	Tension
V	Shear
W	Strain energy of the rod
X	Global X-axis, X-coordinate
Y	Global Y-axis, Y-coordinate
j	Segment number
k	Index in power series expansion
p	Modulus of the elliptic integral
p_{min}	Lower bound of modulus p

p_{max}	Upper bound of modulus p
q	Non-dimensional applied concentrated load
q_n	Non-dimensional applied uniform distributed load normal to the rod
q_y	Non-dimensional applied uniform distributed load along the y-direction
r	Polar coordinate used in describing a cantilever beam
s	Arc length of the rod
w_n	Uniform distributed load normal to the rod
w_t	Uniform distributed load tangent to the rod
w_x	Uniform distributed load along the x-direction
w_y	Uniform distributed load along the y-direction
x	Local x-axis, x-coordinate
y	Local y-axis, y-coordinate
Λ	Non-dimensional moment with respect to the length of the rod
Π	Non-dimensional potential energy with respect to the length of the rod
Φ	The angle of tangent with the local horizontal axis at the end of the rod
Ω	Modulus of the elliptic integral solution for a normal loaded cantilever
Υ	Non-dimensional tension with respect to the length of the rod
β	The angle of tangent with the local horizontal axis at the start of the rod; the angle at location r in a polar coordinate system
α	Subtending angle of arch
γ	The angle of the tangent with the global horizontal axis
δ	Length of one segment
η	Non-dimensional uniform distributed load normal to the rod
θ	The deformed shape of the rod

μ	Non-dimensional moment with respect to the length of a segment
ν	Non-dimensional shear with respect to the length of a segment
ξ	Non-dimensional uniform distributed load tangent to the rod
ρ	Non-dimensional arc length of the rod
τ	Non-dimensional tension with respect to the length of a segment
ϕ	The unstressed shape of the rod
χ	Non-dimensional uniform distributed load along the x-direction; displacement
ψ	Non-dimensional uniform distributed load along the y-direction
$\hat{E}$	Potential energy of the rod
$\overline{\Theta}$	Average rotation of the rod

CHAPTER
ONE

INTRODUCTION

1.1 Background

Considerable attention has been focused on constructing light weight and flexible structures involving high radii of curvature. Applications of such structures can be seen in space exploration, design of mechanical springs, pipeline problems, bridges and other unique problems. These structures can often be treated as thin inextensible rods subject to various boundary conditions. As many of these flexible structures allow large elastic deformations, the conventional small deflection theory is inadequate and the nonlinearity of the problem cannot be neglected.

Finite elastic deflections of bars have been investigated by numerous researchers over the past two centuries and a vast literature can be found in Schmidt and DaDeppo [1], Gorshi and Aust [2] and most recently by Wang [3]. The subject has been studied often because of its inherent nonlinear nature and often because it has application in areas as diverse as the deflections of trees, the characteristics of fabric thread, the laying of pipelines and the deflection of orthodontic appliances. Due to the scope of the subject, it is not the intention of this present investigation to give a complete survey of research done in this area. This survey of literature is in no way exhaustive and is only a limited indication of the available literature pertaining to this subject.

The study of large deformations of rods or bars is generally referred to as elastica

problems. Gorshi and Aust [2] defined elastica as the deflection curve of an elastic bar involving large deformations, however elastica usually implies an inextensible bar with no shear deformation undergoing planar deflections co-planar with the applied loads. The heavy elastica refers to large deflection of elastic bars under their own weight.

The study of the elastica usually begins with the Bernoulli-Euler equation which states that the bending moment at any point along a rod is proportional to the change in curvature of the rod (Frisch-Fay [4]). Since the analysis of a cantilever subject to various loadings is one of the simplest elastica problems, numerous researchers [5,6] have carried out the investigations of this problem (Gorshi and Aust [2]). For example, Frisch-Fay [4] derived the analytical solutions of a cantilever under a free end load using the method of elastic similarity and Mattiasson [7] later numerically computed the solution based on Frisch-Fay's formulation. Moreover, Mitchell [8] derived the closed-form solution to a cantilever subject to a uniform normal load in elliptic integral form by the direct integration method. While the method of solution of the above workers relied on finding the analytical solutions, these are often impossible when the boundary conditions become more complicated. As a result, one must resort to numerical procedures for solving the nonlinear Bernoulli-Euler equations.

Numerical solutions to elastica problem involve solving the corresponding second order nonlinear differential equation either directly or by applying an energy method with variational calculus to arrive at the solution. Conway and Seames [9] presented an iterative method approximating the bar using a number of circular arcs tangent to one another at their points of intersection. The nonlinear problem can thus be linearized in

a way similar to the segmental shooting technique used by Faulkner and Streudlinsky [10]. Surana [11] implemented a nonlinear finite element formulation for analysing two dimensional curved beam elements and he later modified this method to accommodate general three dimensional curved shells [12] and curved beams [13]. This formulation invokes the criterion of minimum potential energy where the tangent stiffness matrix is evaluated using a nonlinear nodal displacement formulation and a Newton Raphson iterative incremental loading path. Also using the energy principle, Somervaille [14] developed a finite element method with an intrinsic coordinate element for computing not only large deflections of elasticae but also for handling related buckling and vibration problems. He then developed higher order transfer matrices [15] solving vibration and column buckling problems.

Stability considerations of structures are always of extreme interest and importance. Elastic stability of curved beams was studied by Lo and Conway [16, 17] who compared the linear and nonlinear extensible theory as well as the nonlinear inextensible theory. Lo and Conway [16, 17] were able to identify the threshold buckling values for a curved beam hinged at fixed supports and subjected to equal but opposite end moments. Furthermore, DaDeppo and Schmidt [18, 19] expressed the large unsymmetric buckling of deep circular arches in elliptic integral form. Such analytical solutions result in a set of nonlinear simultaneous equations which were difficult and laborious to solve. The same difficulty is reported by Huddleston [20] when nonlinear deflections of deep arches were investigated. Finally, Fried [21] utilized variational principles in his finite element formulation to solve the stability of a cantilever under a

free end transverse load. The stability of the resulting deformed configuration is solved as an eigenvalue problem which is a function of the energy tangent stiffness matrix.

For almost all of the studies mentioned, only one equilibrium configuration is obtained for a given load. However, the elastica problem being solved is highly nonlinear and multiple equilibrium solutions are possible as discussed by Fried [21] and more recently by Navaee [22]. Navaee [22] employed examples of a cantilever under different loadings to show the development of multiplicity of solutions as the loading is increased. While the more complicated problems of analysing the stability of arches has been investigated by many researchers [see for example 19, 20, 24-27], the multiplicity of solutions to nonlinear problems has received little attention. This was often due to the numerical difficulties encountered at bifurcation points. As a result, various improved iterative algorithms have been used to extend numerical procedures beyond the bifurcation points [28-30].

It should also be noted that attempts to describe the stability of rods in general terms have been done (see for example [31]). However, this type of analysis is beyond the scope of this thesis.

1.2 Thesis Outline

The general nonlinear bending of rods is described quantatively by the Bernoulli-Euler equation. Three properties of such problems are frequently of interest and importance. These are the deformed configurations of the rod, the associated forces and moments and the stability of the resulting deformed shapes. This current

investigation is divided into five parts with this chapter reviewing and introducing the problem. In Chapter Two, the basic nonlinear differential equations for the bending of rods under concentrated, distributed and normal loading conditions are derived. The solution technique for the resulting nonlinear differential equations is then found by formulating a general numerical procedure, referred to here as the segmental shooting technique. Chapter Three considers the verification of this procedure using several cantilever problems with known solutions. The method of solution is further discussed by examining the multiplicity of the solutions for each problem. In Chapter Four, more complicated problems involving deep circular arches are investigated. The presence of multiple solutions is pursued for each problem. As well, the stability of the deformed configurations is evaluated in a limited sense using the criterion of minimum potential energy. Finally, Chapter Five summarizes the findings as well as discussing the merits and limitations of the numerical techniques used.

CHAPTER
TWO

THEORETICAL FORMULATION

In this chapter, the basic derivation of the governing equilibrium equations for the plane deformations of an inextensible rod are discussed. The general form of such governing equations are presented. In addition, the segmental shooting technique used in this thesis for analysing large deformations of rods subjected to various loading conditions is discussed.

The segmental shooting technique is divided into the segmental approach and the shooting procedure. The segmental approach involves dividing the rod into numerous segments such that the linearization of the governing equations for each segment is possible. The underlying assumption associated with this linearization technique is that the relative rotation between the ends of each individual segment remains small. The technique by which these linearized segments are assembled to represent the non-linear solution of the entire rod is shown. Often, numerous segments, in the order of hundreds, are required to satisfy the above assumption.

An iterative method is employed to convert the original boundary value problem into an initial value problem. The shooting procedure is used to satisfy the boundary conditions by this iterative method. Details of the iterative shooting technique used to solve the initial value problem are illustrated and discussed.

2.1 Equilibrium Equations

Frisch Fay [4] has derived the basic equilibrium equations for the nonlinear deformations of thin rods shown in Figure 2.1. The details of this derivation follow. Let $\phi(s)$ be the angle describing the free or unstressed shape of the rod while $\theta(s)$ is the slope of the deformed configuration of the rod at position s along the arc length. It is assumed that the rod is inextensible so that the deformation is strictly bending. It is also assumed that the rod is governed by the Bernoulli-Euler equation which means that the bending moment M at any point in the rod is proportional to the change in curvature of the rod,

$$M = EI\left[\frac{d\theta}{ds} - \frac{d\phi}{ds}\right], \tag{2.1}$$

Here EI is the flexural rigidity of the rod and

$$\frac{d\phi}{ds} = \frac{1}{R(s)}, \tag{2.2}$$

where the initial radius of curvature of the free shape is R(s).

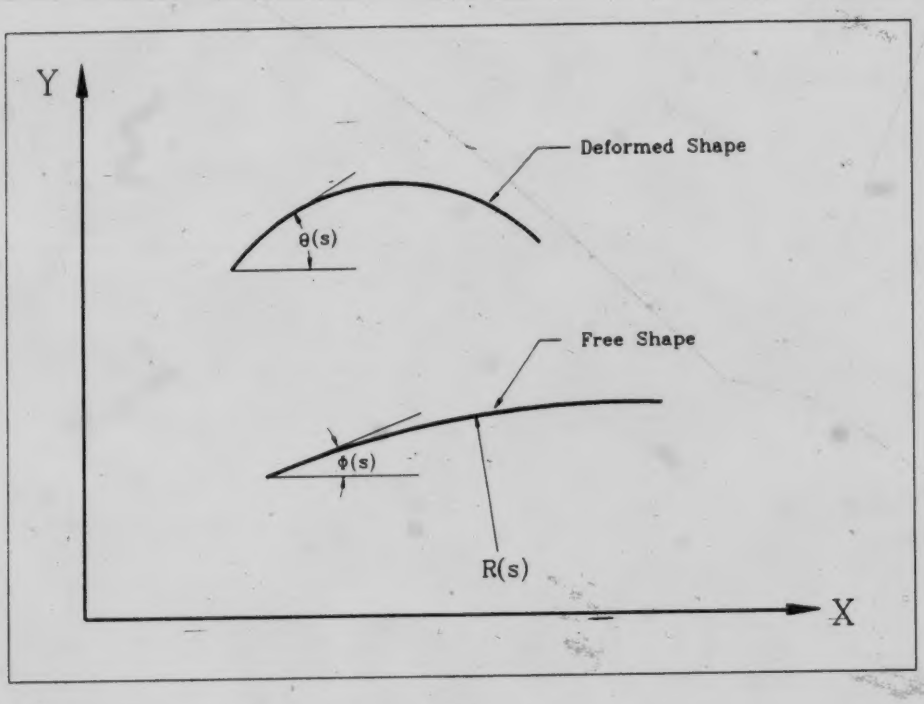


Figure 2.1 Free and Deformed Shape of the Rod

Now consider an infinitesimal element of the rod of length *d*s shown in Figure 2.2. This element has tension T, shear V, bending moment M, which can vary along its length as well as horizontal and vertical distributed loads w_x and w_y. Furthermore, the directions of the forces and moments shown in Figure 2.2 are positive using the sign conventions adopted in this work.

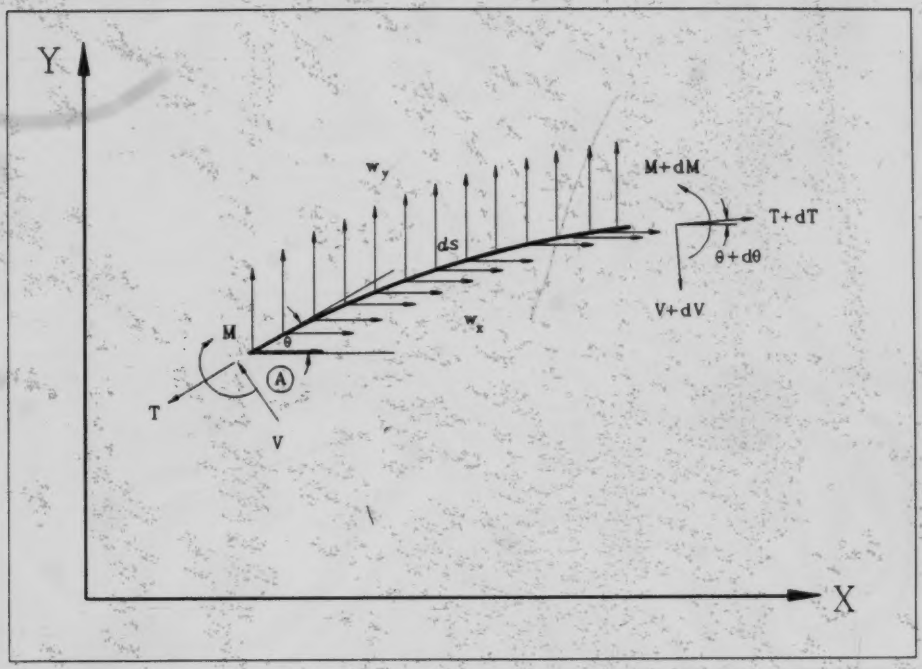


Figure 2.2 Infinitesimal Element Loaded by Horizontal and Vertical Distributed Load

Static equilibrium conditions on the element results in

$$\rightarrow+ \sum F_x = 0 = (T+dT)\cos(\theta+d\theta) + (V+dV)\sin(\theta+d\theta) - T\cos\theta - V\sin\theta + w_x ds, \tag{2.3}$$

$$+\uparrow \sum F_y = 0 = (T+dT)\sin(\theta+d\theta) - (V+dV)\cos(\theta+d\theta) - T\sin\theta + V\cos\theta + w_y ds, \tag{2.4}$$

$$+ \sum M_A = 0 = (V+dV)(\cos\theta)ds + (T+dT)(\sin\theta)ds + M - (M+dM) + w_x \sin\theta ds\frac{ds}{2} - w_y \cos\theta ds\frac{ds}{2}, \tag{2.5}$$

leading to

$$\frac{d}{ds}(T\cos\theta) + \frac{d}{ds}(V\sin\theta) + w_x = 0, \tag{2.6}$$

$$\frac{d}{ds}(T\sin\theta) - \frac{d}{ds}(V\cos\theta) - w_y = 0. \tag{2.7}$$

The moment equilibrium yields

$$V = \frac{dM}{ds}. \tag{2.8}$$

Integrating equations (2.6) and (2.7) over the rod with length s gives,

$$T\cos\theta + V\sin\theta + \int_0^s w_x ds = C_1 \tag{2.9}$$

and

$$T\sin\theta - V\cos\theta + \int_0^s w_y ds = C_2. \tag{2.10}$$

where C_1 and C_2 are constants of integration. These equations can be combined by multiplying equation (2.9) by $\sin\theta$ and adding to equation (2.10) multiplied by $-\cos\theta$ to yield

$$V = C_1\sin\theta - C_2\cos\theta - \sin\theta\int_0^s w_x\,ds + \cos\theta\int_0^s w_y\,ds. \qquad (2.11)$$

Using equations (2.1), (2.2) and (2.8), equation (2.11) becomes

$$\frac{d}{ds}\left[EI\left(\frac{d\theta}{ds} - \frac{1}{R(s)}\right)\right] = C_1\sin\theta - C_2\cos\theta - \sin\theta\int_0^s w_x\,ds + \cos\theta\int_0^s w_x\,ds. \qquad (2.12)$$

Instead of using x and y components for the distributed forces, it is often convenient to express them in terms of normal and tangential distributed loads, w_n and w_t respectively, as shown in Figure 2.3.

Since

$$w_x = w_t\cos\theta + w_n\sin\theta \qquad (2.13)$$

and

$$w_y = w_t\sin\theta - w_n\cos\theta, \qquad (2.14)$$

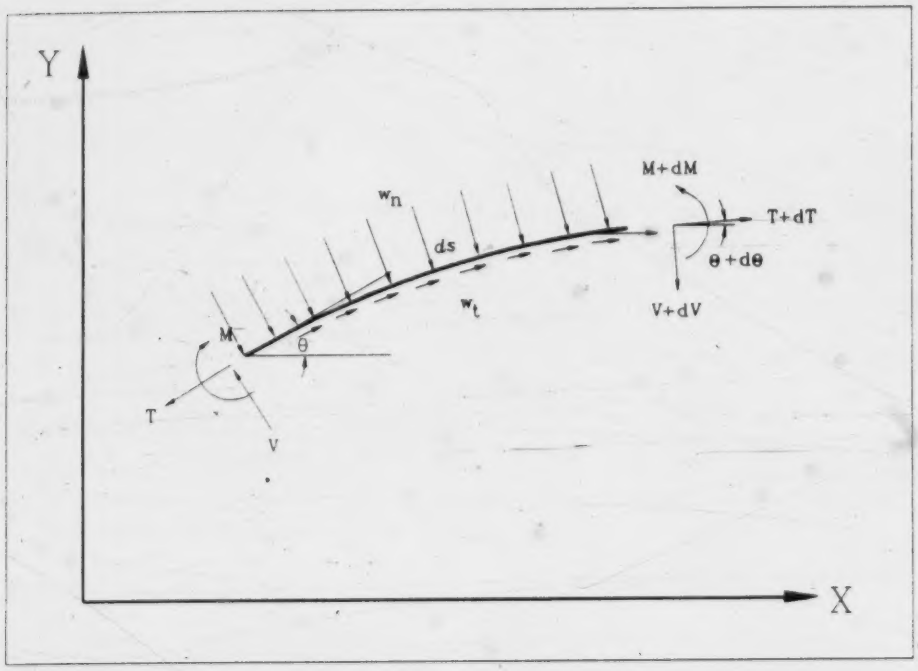


Figure 2.3 Infinitesimal Element Loaded by Normal and Tangential Distributed Load

equation (2.12) can be written as

$$\frac{d}{ds}[EI(\frac{d\theta}{ds}-\frac{1}{R(s)})] = C_1\sin\theta - C_2\cos\theta$$
$$-\sin\theta\int_0^s(w_t\cos\theta + w_n\sin\theta)ds + \cos\theta\int_0^s(w_t\sin\theta - w_n\cos\theta)ds. \qquad (2.15)$$

If the flexural rigidity EI, radius of curvature R and prescribed distributed load w_x, w_y or w_n, w_t are constant throughout a rod element, equations (2.12) and (2.15) simplify to

$$EI\frac{d^2\theta}{ds^2} = C_1\sin\theta - C_2\cos\theta - sw_x\sin\theta + sw_y\cos\theta \tag{2.16}$$

and

$$\begin{aligned} EI\frac{d^2\theta}{ds^2} = {} & C_1\sin\theta - C_2\cos\theta - w_t\sin\theta\int_0^s\cos\theta ds - w_n\sin\theta\int_0^s\sin\theta ds \\ & + w_t\cos\theta\int_0^s\sin\theta ds - w_n\cos\theta\int_0^s\cos\theta ds \end{aligned} \tag{2.17}$$

respectively. Equations (2.16) and (2.17) are the second order nonlinear differential equations describing the deformed geometry $\theta(s)$ of the rod. The constants of integration C_1 and C_2 are evaluated from the boundary conditions. Applying boundary conditions $T = T_1$, $V = V_1$ and $\theta = \beta$ at $s = 0$, which are the initial values of tension shear and slope of the rod, equations (2.9) and (2.10) become

$$C_1 = T_1\cos\beta + V_1\sin\beta, \tag{2.18}$$

$$C_2 = T_1\sin\beta - V_1\cos\beta. \tag{2.19}$$

Equations (2.16) and (2.17) along with (2.18) and (2.19) determine the deformed shape $\theta(s)$ of the rod subjected to prescribed distributed loading. If the rotation, θ, of a rod or rod segment is small relative to a suitable local coordinate system, equations (2.16) and (2.17) can be linearized by writing $\sin\theta \approx \theta$ and $\cos\theta \approx 1$. Discussion of this local

coordinate system is deferred until later. With this approximation, equation (2.16) becomes

$$EI\frac{d^2\theta}{ds^2} = C_1\theta - C_2 - sw_x\theta + sw_y. \tag{2.20}$$

Neglecting small terms of order θ^2 and using linearized forms of equations (2.18) and (2.19) for the constants of integration results in

$$EI\frac{d^2\theta}{ds^2} = T_1(\theta - \beta) + V_1 - sw_x\theta + sw_y, \tag{2.21}$$

where $\theta = \beta$ at $s = 0$. Similarly, equation (2.17) becomes

$$EI\frac{d^2\theta}{ds^2} = T_1(\theta - \beta) + V_1 - w_t\theta s + w_n\theta\int_0^s \theta ds + w_t\int_0^s \theta ds - w_n s. \tag{2.22}$$

Define

$$\bar{\theta}s = \int_0^s \theta ds, \tag{2.23}$$

where Θ represents an average rotation, equation (2.22) becomes

$$EI\frac{d^2\theta}{ds^2} = T_1(\theta - \beta) + V_1 + w_t(\bar{\theta} - \theta)s - w_n(\theta\bar{\theta} + 1)s. \tag{2.24}$$

Neglecting the higher order terms finally results in

$$EI\frac{d^2\theta}{ds^2} = T_1(\theta-\beta)+V_1+w_t(\bar{\theta}-\theta)s-w_n s \tag{2.25}$$

which is the linearized form of equation (2.17).

Introducing the following dimensionless parameters,

$$\begin{aligned}
&\rho = \frac{s}{\delta},\quad \tau = \frac{T\delta^2}{EI},\quad \nu = \frac{V\delta^2}{EI},\\
&\mu = \frac{M\delta}{EI} = \frac{\delta}{EI}\left[EI\left(\frac{d\theta}{ds}-\frac{1}{R}\right)\right] = \left(\frac{d\theta}{d\rho}-\frac{\delta}{R}\right),\\
&\eta = \frac{w_n\delta^3}{EI},\quad \xi = \frac{w_t\delta^3}{EI},\quad \chi = \frac{w_x\delta^3}{EI},\quad \psi = \frac{w_y\delta^3}{EI},
\end{aligned} \tag{2.26}$$

the dimensionless form of equations (2.21) and (2.25) become respectively,

$$\frac{d^2\theta}{d\rho^2} = \tau_1(\theta-\beta)+\nu_1-\rho\chi\theta+\rho\psi, \tag{2.27}$$

and

$$\frac{d^2\theta}{d\rho^2} = \tau_1(\theta-\beta)+\nu_1+\xi\rho(\bar{\theta}-\theta)-\eta\rho. \tag{2.28}$$

The solution of these equations is achieved using a power series along with appropriate boundary conditions as follows

$$\theta = \sum_{k=0}^{\infty} \alpha_k \rho^k, \tag{2.29}$$

$$\theta|_{\rho=0} = \beta, \tag{2.30}$$

$$\frac{d\theta}{d\rho}|_{\rho=0} = \mu_1 + \frac{\delta}{R}, \tag{2.31}$$

$$\frac{d^2\theta}{d\rho^2}|_{\rho=0} = \nu_1. \tag{2.32}$$

For the rod loaded by uniform horizontal and vertical loads then

$$\frac{d^3\theta}{d\rho^3}|_{\rho=0} = \tau_1(\mu_1 + \frac{\delta}{R}) - \chi\beta + \psi. \tag{2.33}$$

Similarly, if the rod is loaded by uniform normal and tangential loads then

$$\frac{d^3\theta}{d\rho^3}|_{\rho=0} = \tau_1(\mu_1 + \frac{\delta}{R}) - \eta. \tag{2.34}$$

Substituting equations (2.30), (2.31), (2.32), and (2.33) or (2.34) into (2.27) and (2.28) results in the coefficients α_k being given by

$$\alpha_0 = \beta, \tag{2.35}$$

$$\alpha_1 = \mu_1 + \frac{\delta}{R}, \tag{2.36}$$

$$\alpha_2 = \frac{\nu_1}{2}, \tag{2.37}$$

$$\alpha_3 = \frac{(\tau_1\alpha_1 - \chi\beta - \psi)}{6}, \tag{2.38}$$

or

$$\alpha_3 = \frac{(\tau_1\alpha_1 - \eta)}{6}, \tag{2.39}$$

and

$$\alpha_k = \frac{\tau_1\alpha_{k-2} - \chi\alpha_{k-3}}{k(k-1)} \qquad k = 4,5,6,\ldots\infty, \tag{2.40}$$

or

$$\alpha_k = \frac{\tau_1\alpha_{k-2} - \xi\alpha_{k-3}(1 - \frac{1}{k-2})}{k(k-1)} \qquad k = 4,5,6,\ldots\infty. \tag{2.41}$$

Equations (2.38) and (2.40) are the coefficients of the power series in terms of distributed horizontal and vertical loads w_x, w_y while equations (2.39) and (2.41) are expressed as a function of distributed normal and tangential loads w_n, w_t to the rod

segment. Equations (2.29) and (2.35) through (2.41) give the solution for a linearized rod segment in terms of the values of the forces, moments and distributed load at the beginning of the rod segment.

2.2 Segmental Technique

In the previous section, the basic equilibrium equations of a rod undergoing large deformations are formulated. The resulting nonlinear second order differential equations are linearized by assuming a small relative rotation between the ends of the rod. However, many problems involve large deformations resulting in large relative rotations between the ends of the rod. As a result, in such a case the previous assumption that $\sin\theta \approx \theta$ and $\cos\theta \approx 1$ are invalid. However, if the rod is divided into numerous segments, each segment having a small relative rotation between the ends, the linearized solution can then be used. A linear second order differential equation has the advantage of being easily solved by standard numerical techniques. Large deformations of the rod can thus be computed easily by assembling a sequence of linearized solutions for each segment in a continuous manner.

This numerical procedure, referred to here as the segmental technique, was developed and first applied to problems associated with laying of offshore pipelines by Faulkner and Stredulinsky [10] and later modified by El-Rayes [33] to problems associated with investigating forces and moments generated by orthodontic appliances. Both investigations involve structures that differ vastly in physical size but the same technique was used to analyze nonlinear problems associated with large deflections. The

subsequent derivations of this numerical procedure closely follow their works.

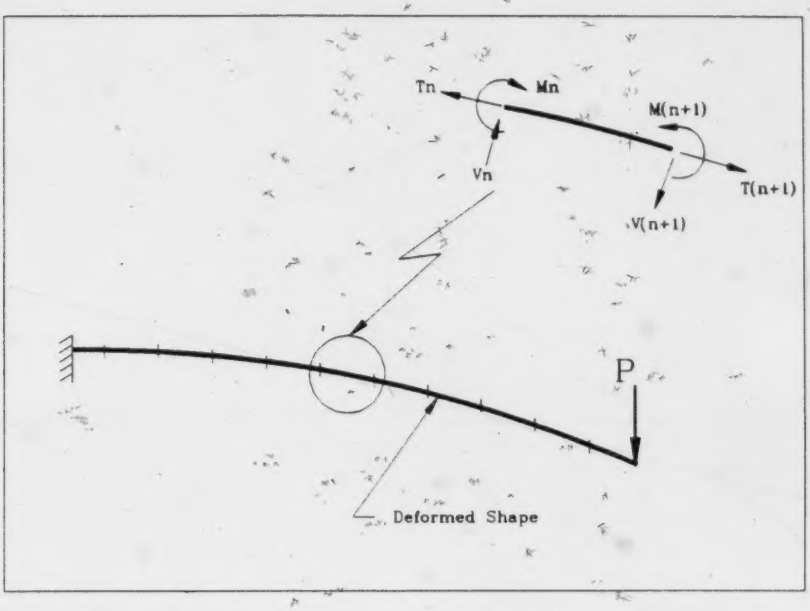


Figure 2.4 Application of the Segmental Approach

In order to illustrate the linearization technique of the segmental approach, consider as an example a cantilever loaded at the free end as shown in Figure 2.4. The load P applied is sufficiently large such that the cantilever undergoes nonlinear deflections which are governed by the previously derived second order nonlinear differential equation. The cantilever is then divided into segments with the relative rotation between the ends of each segment being small. Each segment has its initial values as shear V_n, tension T_n and moment M_n. The linearization technique allows computing of the corresponding shear V_{n+1}, tension T_{n+1} and moment M_{n+1} at the end of a segment. While maintaining continuity of force and geometry between segments, this process is repeated for all segments forming the rod. In other words, the segmental

approach essentially solves a nonlinear rod problem by computing and assembling a sequence of linearized rod problems. Therefore, the overall deflections of the rod can be large and linearized solutions can still be used given a sufficient number of segments.

Consider a segment of the rod of finite length δ as shown in Figure 2.5. In order to employ the linear solution to the segment, a coordinate system x and y is defined to ensure that the relative rotations between the two ends of the segment is small. As previously stated the number of segments used is chosen so that θ, the angle with respect to the local coordinate system, remains small over any one segment.

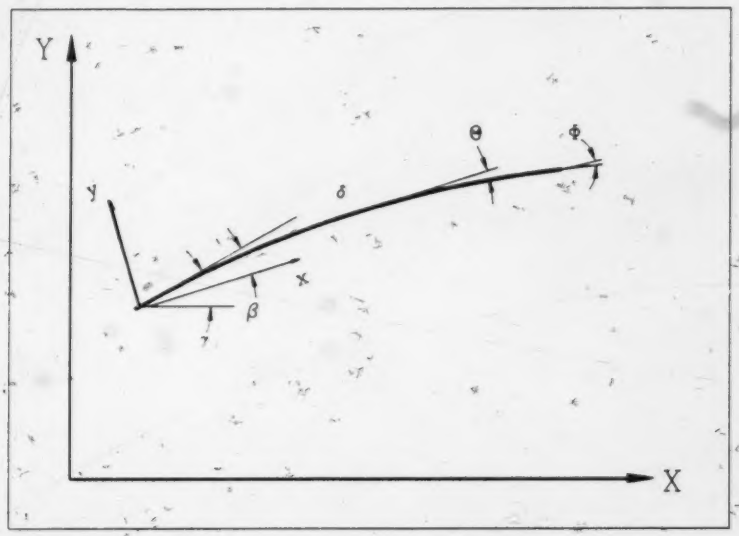


Figure 2.5 Local Coordinate System of an Element

Here γ is the tangent angle of the segment with respect to the global X-axis. β and Φ are respectively the tangent at the start and end of the rod with respect to the local x-axis. The use of the power series expansion to solve the linearized equation numerically requires the problem to be treated as an initial value problem instead of a boundary value

problem. That is, complete knowledge is required at the start of the rod. Since the segmental approach is examining the rod as a series of small segments, the problem can be handled as a series of initial value problems on a segment to segment basis. The solution of the rod can then be constructed by preserving force and geometric compatibility between segments as shown in Figure 2.6 for the case of a cantilever with end load. The problem thus requires solving a linear differential equation for each segment in which the following six initial conditions are required;

X_1	Global X Coordinate at Start of Segment
Y_1	Global Y Coordinate at Start of Segment
γ_1	Angle of Tangent with Global X axis at Start of Segment
T_1	Start node Tension Value at Start of Segment
V_1	Start node Shear Value at Start of Segment
M_1	Start node Moment Value at Start of Segment

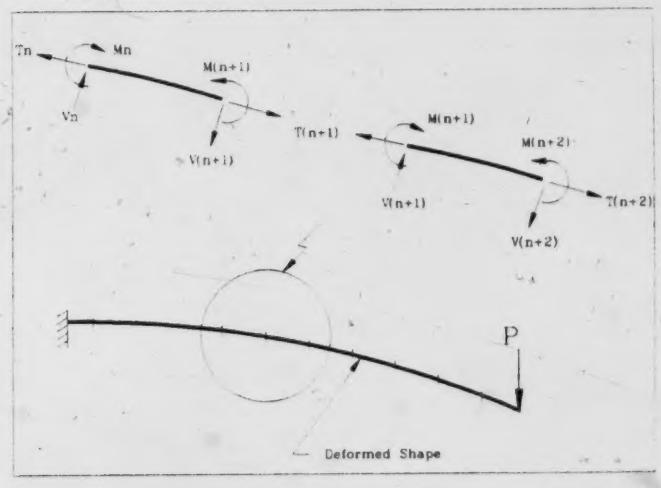


Figure 2.6 Numerous Linear Segments Assembled to Form a Nonlinear Problem

Given all of the above six initial conditions, the corresponding force and geometric properties at the end of each segment are computed. Finally, the six variables at the end of the rod are determined by appropriately assembling the segments together.

Figures 2.7 and 2.8 depict the assembling of successive segments. Figure 2.7 shows the geometric compatibility while Figure 2.8 demonstrates the force compatibility of the segments. Segment j and $(j + 1)$ are shown with nodes $(i - 1)$, i and $(i + 1)$. Each segment consists of two nodes and node i is common to both segments. For simplicity of illustration, distributed loading is omitted. Six parameters are defined at each node as shown in Table 2.1.

Table 2.1 Nodal Connectivity between Successive Segments

NODE/SEGMENT	j	$j + 1$
$i - 1$	$X_{1(j)}\ Y_{1(j)}\ \gamma_{1(j)}$ $T_{1(j)}\ V_{1(j)}\ M_{1(j)}$	
i	$X_{2(j)}\ Y_{2(j)}\ \gamma_{2(j)}$ $T_{2(j)}\ V_{2(j)}\ M_{2(j)}$	$X_{1(j+1)}\ Y_{1(j+1)}\ \gamma_{1(j+1)}$ $T_{1(j+1)}\ V_{1(j+1)}\ M_{1(j+1)}$
$i + 1$		$X_{2(j+1)}\ Y_{2(j+1)}\ \gamma_{2(j+1)}$ $T_{2(j+1)}\ V_{2(j+1)}\ M_{2(j+1)}$

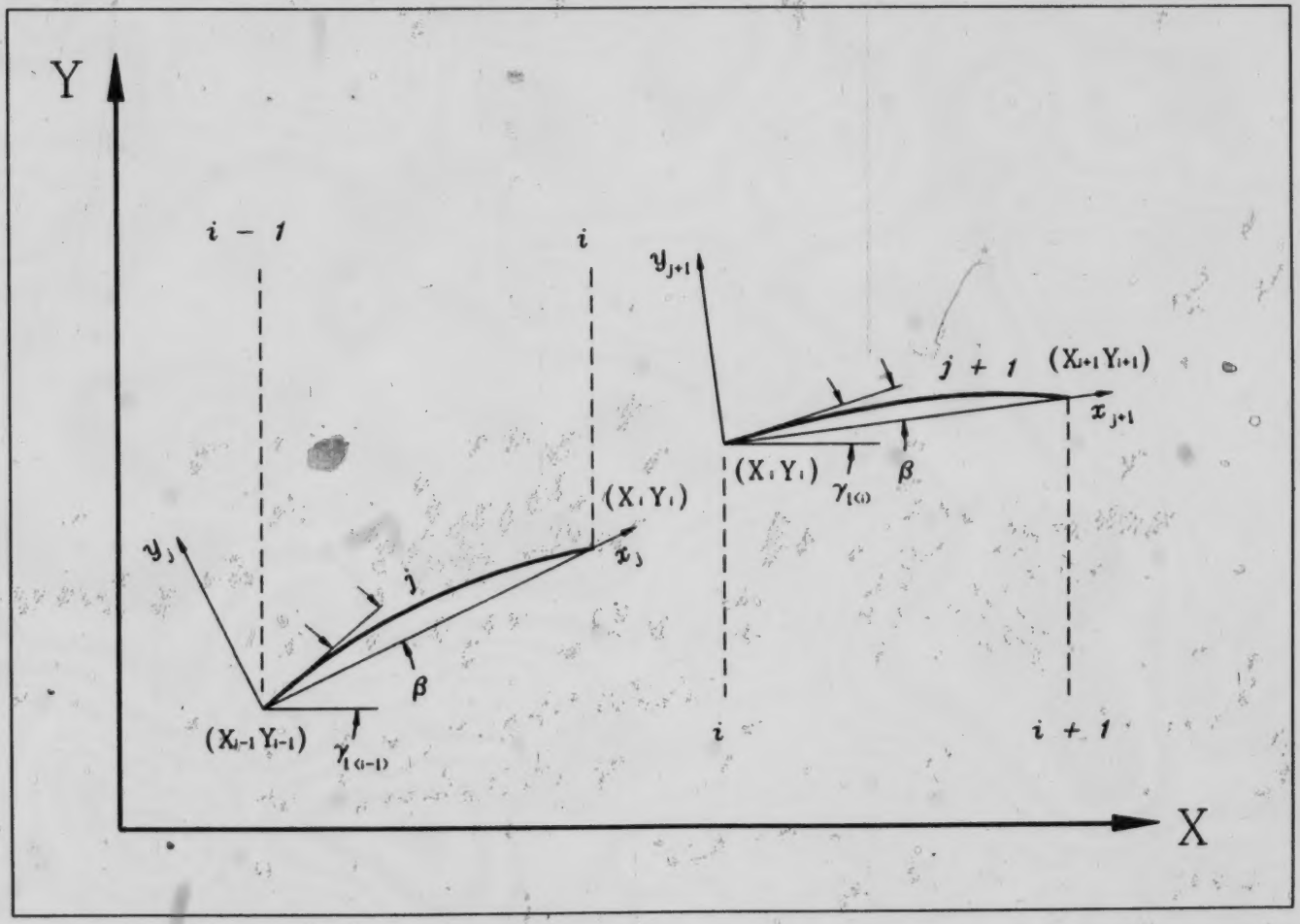


Figure 2.7 Geometric Compatibility of Successive Segments

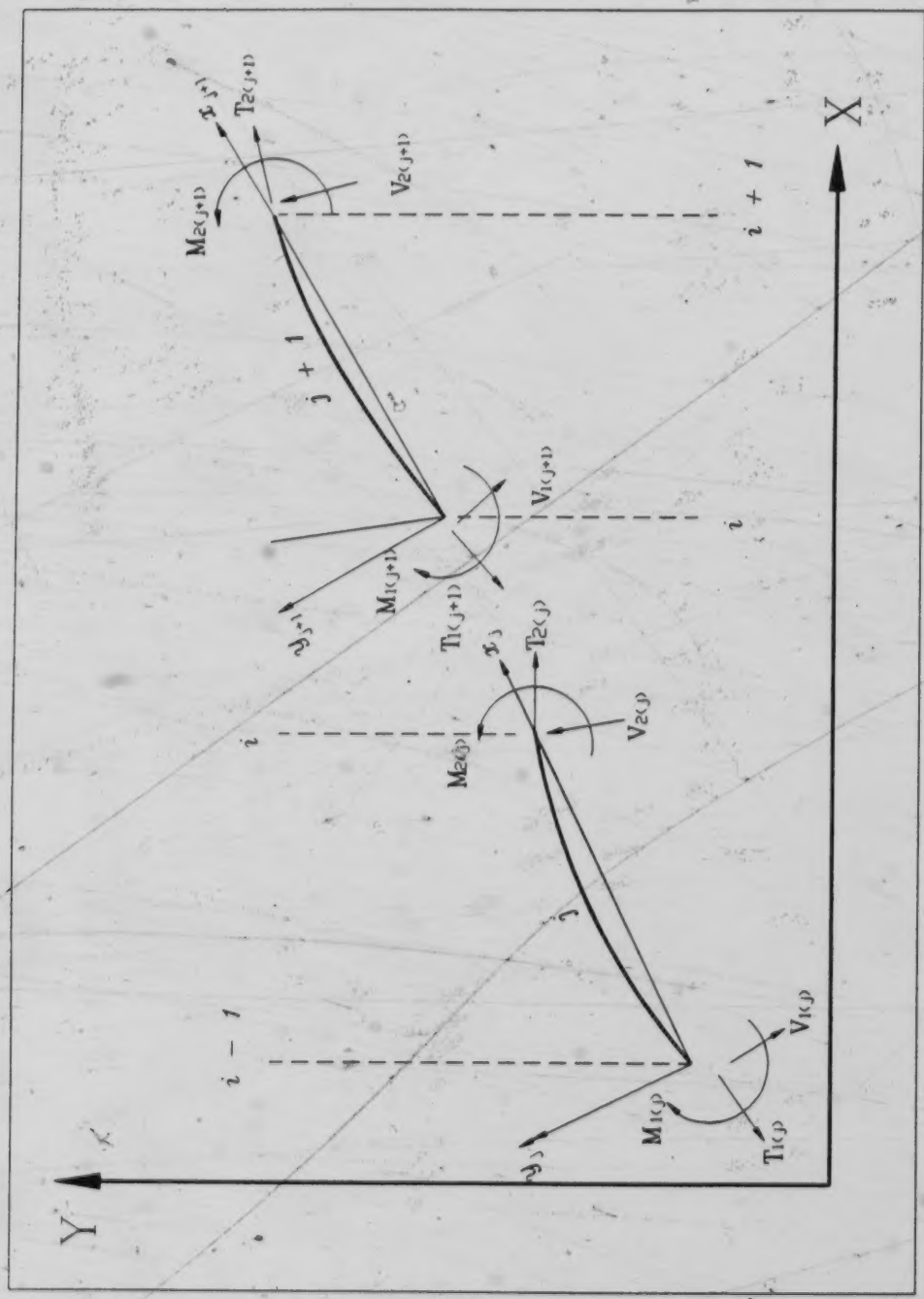


Figure 2.8 Force Compatibility of Successive Segments

Continuity between segments is achieved by equating

$$\begin{matrix} T_{2(i)} = T_{1(i+1)}, & V_{2(i)} = V_{1(i+1)}, & M_{2(i)} = M_{1(i+1)} \\ X_{2(i)} = X_{1(i+1)}, & Y_{2(i)} = Y_{1(i+1)}, & \gamma_{2(i)} = \gamma_{1(i+1)} \end{matrix} \tag{2.42}$$

Six parameters $T_{1(j)}$, $V_{1(j)}$, $M_{1(j)}$, $X_{1(j)}$, $Y_{1(j)}$ and $\gamma_{1(j)}$ are required at node $(i - 1)$ to compute $T_{2(j)}$, $V_{2(j)}$, $M_{2(j)}$, $X_{2(j)}$, $Y_{2(j)}$ and $\gamma_{2(j)}$ for element j. Then the six parameters at node i for segment j are equated to those for segment $j + 1$ using equation (2.42). This sequence of computation is repeated for all segments used to define the rod. The flexural rigidity EI, radius of curvature R and the length δ of each segment allows the non-dimensionalizing of the six specified parameters as given in equation (2.26). Therefore,

$$\mu_1 = \frac{M_1 \delta}{EI}, \tag{2.43}$$

$$\tau_1 = \frac{T_1 \delta^2}{EI}, \tag{2.44}$$

$$\nu_1 = \frac{V_1 \delta^2}{EI}. \tag{2.45}$$

As mentioned previously, the relative rotation θ within each segment is assumed to be small, and therefore a local coordinate system (x,y) is selected for each segment so that $|\theta| << 1$ is guaranteed. The orientation of the coordinate system (x,y) is optimized by selecting the initial slope, β, so that $(\theta^2)_{AVG}$ is kept approximately minimum. Recall

that the deformed shape of the element is given by the power series expansion as,

$$\theta = \beta + (\mu_1 + \frac{\delta}{R})\rho + \frac{\nu_1}{2}\rho^2 + \frac{(\tau_1\alpha_1 - \chi\beta - \psi)}{6}\rho^3 + \\ + \ldots + \frac{(\tau_1\alpha_{k-2} - \chi\alpha_{k-3})}{k(k-1)}\rho^k + \ldots \\ k = 4,5,6,\ldots\infty \qquad (2.46)$$

or

$$\theta = \beta + (\mu_1 + \frac{\delta}{EI})\rho + \frac{\nu_1}{2}\rho^2 + \frac{\tau_1\alpha_1 - \eta}{6}\rho^3 + \ldots \\ + \frac{(\tau_1\alpha_{k-2} - \xi\alpha_{k-3}(1 - \frac{1}{k}))}{k(k-1)}\rho^k + \ldots \\ k = 4,5,6,\ldots\infty \qquad (2.47)$$

depending whether horizontal/vertical or normal/tangential loading is specified. The first three terms are equal for both equations and the fourth term is in the order of ρ^3 where $0 \leq \rho \leq 1$. Consequently, θ can be approximated by considering the first three terms of either series,

$$\theta = \beta + (\mu_1 + \frac{\delta}{R})\rho + \frac{\nu_1}{2}\rho^2. \qquad (2.48)$$

The average of θ^2 over the length of the segment is given by

$$\theta^2_{AVG} = \int_0^1 \theta^2 d\rho = \int_0^1 (\beta^2 + 2\beta(\mu_1 + \frac{\delta}{R})\rho + (\nu_1\beta + \mu_1^2 + 2\mu_1\frac{\delta}{R} + \frac{\delta^2}{R})\rho^2 + \nu_1(\mu_1 + \frac{\delta}{R})\rho^3 + \frac{\nu_1^2}{4}\rho^4)d\rho \quad (2.49)$$

$$= \beta^2 + \beta(\mu_1 + \frac{\delta}{R}) + \frac{1}{3}(\nu_1\beta + \mu_1^2 + 2\mu_1\frac{\delta}{R} + \frac{\delta^2}{R}) + \frac{\nu_1}{4}(\mu_1 + \frac{\delta}{R}) + \frac{\nu_1^2}{20}.$$

Minimizing $(\theta^2)_{AVG}$ with respect to β,

$$\frac{d\theta^2_{AVG}}{d\beta} = 2\beta + (\mu_1 + \frac{\delta}{R} + \frac{\nu_1}{3}) = 0 \quad (2.50)$$

yields

$$\beta = -\frac{1}{2}(\mu_1 + \frac{\delta}{R} + \frac{\nu_1}{3}). \quad (2.51)$$

Therefore, the local coordinate system (x,y) is located at the start of each segment using β from equation (2.51) as shown both in Figures 2.5 and 2.7.

In addition, the transformation between the global and local coordinate system is required to maintain overall continuity. The slope at the end of the segment with respect to the local axis is given by,

$$\Phi = \theta|_{\rho-1} - \sum_{k=0}^{\infty} \alpha_k \rho^k|_{\rho-1} = \sum_{k=0}^{\infty} \alpha_k \tag{2.52}$$

As shown in Figure 2.7, $\gamma_{(i-1)}$ and γ_i are the tangent of the angle between the start and end point of the segment with the global X axis. Therefore, the end point of the segment would make an angle with respect to the global X axis given by

$$\gamma_i = \gamma_{i-1} - \beta_j + \Phi_j \tag{2.53}$$

On a per segment basis, the local coordinates of the end node are given by using geometric relations,

$$x = \int_0^s \cos\theta ds = \int_0^s ds = s, \quad \frac{x}{\delta} = \rho, \tag{2.54}$$

and

$$y = \int_o^s \sin\theta ds = \int_0^s \theta ds \Rightarrow \frac{y}{\delta} = \int_0^s \frac{\theta}{\delta} ds = \int_0^\rho \theta d\rho = \sum_{k=0}^{\infty} \frac{\alpha_k \rho^{k+1}}{k+1} \tag{2.55}$$

Since six parameters are specified at the start node (i - 1) of a segment, it is possible to express the parameters at end node (i) as a function of the start node parameters. Using geometric relationships, the nodal points of a segment can be resolved relative to the global X and Y coordinate.

$$X_i = X_{i-1} + x_2\cos(\gamma_{i-1} - \beta) - y_2\sin(\gamma_{i-1} - \beta), \tag{2.56}$$

$$Y_i = Y_{i-1} + x_2\sin(\gamma_{i-1} - \beta) + y_2\cos(\gamma_{i-1} - \beta) \tag{2.57}$$

and thus global continuity can be maintained. At the end point of a segment (i.e., $\rho = 1$), equation (2.55) becomes

$$\frac{y_2}{\delta} = \sum_{k=0}^{\infty} \frac{\alpha_k}{k+1} \tag{2.58}$$

and equations (2.56) and (2.57) simplify to

$$X_i = X_{i-1} + \delta\cos(\gamma_{i-1} - \beta) - \delta h_j\sin(\gamma_{i-1} - \beta), \tag{2.59}$$

$$Y_i = Y_{i-1} + \delta\sin(\gamma_{i-1} - \beta) + \delta h_j\cos(\gamma_{i-1} - \beta). \tag{2.60}$$

Recall that the dimensionless bending moment and shear are respectively

$$\mu = \left(\frac{d\theta}{d\rho} - \frac{\delta}{R}\right) = \left(\sum_{k=1}^{\infty} (k\alpha_k\rho^{k-1}) - \frac{\delta}{R}\right) \tag{2.61}$$

and

$$\nu = \frac{d}{d\rho}\left(\frac{d\theta}{d\rho} - \frac{\delta}{R}\right) = \sum_{k=2}^{\infty} k(k-1)\alpha_k\rho^{k-2}. \tag{2.62}$$

The bending moment μ and shear υ at the end of a segment (i.e., at $\rho = 1$) can thus be written as

$$\mu_2 = \sum_{k=1}^{\infty} (k\alpha_k) - \frac{\delta}{R}, \tag{2.63}$$

$$\nu_2 = \sum_{k=2}^{\infty} k(k-1)\alpha_k \tag{2.64}$$

The dimensionless tension force at the end of a segment is computed by considering equilibrium along the local x-axis.

$$+\rightarrow \sum f_x = 0 = \tau_2 \cos\Phi + \nu_2 \sin\Phi - \tau_1 \cos\beta - \nu_1 \sin\beta + \chi, \tag{2.65}$$

or

$$+\rightarrow \sum f_x = 0 = \tau_2 \cos\Phi + \nu_2 \sin\Phi - \tau_1 \cos\beta - \nu_1 \sin\beta + \xi + \int_0^1 \sin\theta d\rho \tag{2.66}$$

Equations (2.65) and (2.66) represent static equilibrium along the local x-axis of the rod under horizontal/vertical and normal/tangential distributed load respectively and again using the fact that the relative rotation between the start and end of a segment is small, these simplify to

$$\tau_2 = \tau_1 + \nu_1\beta - \nu_2\Phi - \chi. \tag{2.67}$$

$$\tau_2 = \tau_1 + \nu_1\beta - \mu_2\Phi - \zeta - \eta h_j. \tag{2.68}$$

Consequently, the corresponding moment, shear and tension at the end of a segment are

$$M_2 = \frac{\mu_2 EI}{\delta}, \tag{2.69}$$

$$V_2 = \frac{\nu_2 EI}{\delta^2}, \tag{2.70}$$

$$T_2 = \frac{\tau_2 EI}{\delta^2}. \tag{2.71}$$

Given the forces, slope and position of the start node of a segment, the corresponding values at the end of the segment can be determined. The continuity of the rod is maintained by referencing the local segment coordinates (x,y) with the global coordinates (X,Y). Therefore, the successive segments throughout the rod are assembled based on geometric and force compatibility.

2.3 Shooting Method

If X_1 Y_1, γ_1, M_1, V_1 and T_1 for the first segment are given *a priori* then the segmental technique is able to provide the corresponding values at the end of the rod. A false position Newton Raphson method (*regula falsi*) is employed to correct or improve

the assumed values(s) at the start.

Assuming that there are N unknowns at the start node, then N knowns are required at the end. Let $\underline{A}$ be a vector containing the initial values for the unknowns of the initial node. Let $\underline{B}$ and $\underline{C}$ respectively represent the vector with the known boundary values and the computed boundary values at the end node respectively.

That is,

$$\underline{A}_{(k)} = \begin{pmatrix} A_{1(k)} \\ A_{2(k)} \\ . \\ . \\ . \\ A_{N(k)} \end{pmatrix}, \quad \underline{B}_{(k)} = \begin{pmatrix} B_{1(k)} \\ B_{2(k)} \\ . \\ . \\ . \\ B_{N(k)} \end{pmatrix}, \quad \underline{C}_{(k)} = \begin{pmatrix} C_{1(k)} \\ C_{2(k)} \\ . \\ . \\ . \\ C_{N(k)} \end{pmatrix}. \qquad (2.72)$$

Now the error function $\underline{E}_{(k)} = \underline{B}_{(k)} - \underline{C}_{(k)}$, is defined where k is an iteration counter so that

$$\underline{E}_{(k)} = f(\underline{A}_{(k)}) = f(A_{1(k)}, A_{2(k)}, \ldots A_{N(k)}). \qquad (2.73)$$

The functional dependency of the error vector on the initial unknown values follows because the computed values at the end of the rod depend on the values assumed at the start of the rod. The objective is to find $\underline{A}_{(k)}$ such that $f(\underline{A}_{(k)}) = \underline{E}_{(k)} \approx \underline{0}$. Therefore, the problem requires solving a set of N non-linear simultaneous equations in the form,

$$
\begin{aligned}
E_{1(k)} &= f_1(A_{1(k)}, A_{2(k)}, \ldots A_{N(k)}) = 0 \\
E_{2(k)} &= f_2(A_{1(k)}, A_{2(k)}, \ldots A_{N(k)}) = 0 \\
&\vdots \\
E_{N(k)} &= f_N(A_{1(k)}, A_{2(k)}, \ldots A_{N(k)}) = 0
\end{aligned}
\tag{2.74}
$$

The usual Newton Raphson method for many variables is based on a first order Taylor series [32] which when expanded in matrix form yields

$$
\begin{Bmatrix} E_{1(k+1)} \\ E_{2(k+1)} \\ \vdots \\ E_{N(k+1)} \end{Bmatrix} = \begin{Bmatrix} E_{1(k)} \\ E_{2(k)} \\ \vdots \\ E_{N(k)} \end{Bmatrix} + \begin{bmatrix} \frac{\partial E_{1(k)}}{\partial A_{1(k)}} & \frac{\partial E_{1(k)}}{\partial A_{2(k)}} & \cdots & \frac{\partial E_{1(k)}}{\partial A_{N(k)}} \\ \frac{\partial E_{2(k)}}{\partial A_{1(k)}} & \frac{\partial E_{2(k)}}{\partial A_{2(k)}} & \cdots & \frac{\partial E_{2(k)}}{\partial A_{N(k)}} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial E_{N(k)}}{\partial A_{1(k)}} & \frac{\partial E_{N(k)}}{\partial A_{2(k)}} & \cdots & \frac{\partial E_{N(k)}}{\partial A_{N(k)}} \end{bmatrix} \begin{Bmatrix} A_{1(k+1)} - A_{1(k)} \\ A_{2(k+1)} - A_{2(k)} \\ \vdots \\ A_{N(k+1)} - A_{N(k)} \end{Bmatrix} \tag{2.75}
$$

or

$$
\Rightarrow \underline{E}_{k+1} = \underline{E}_{(k)} + [J_{(k)}](\underline{A}_{(k+1)} - \underline{A}_{(k)}). \tag{2.76}
$$

Setting $\underline{E}_{(k+1)} = \underline{0}$ and solving for $\underline{A}_{(k+1)}$, which are the new or improved estimates of the initial unknowns at the beginning of the rod results in

$$\underline{A}_{(k+1)} = \underline{A}_{(k)} - [\underline{J}_{(k)}]^{-1}\underline{E}_{(k)}. \tag{2.77}$$

Since $\underline{E}$ is not an analytical function, the elements in the Jacobian $[\underline{J}]$ can only be evaluated numerically. A false position backward difference formulation is used to evaluate the partial derivatives of $\underline{E}$ with respect to $\underline{A}$. Two initial guesses per unknown are thus required.

$$\mathbf{J}_{ij(k)} = \frac{\partial \mathbf{E}_{i(k)}}{\partial \mathbf{A}_{j(k)}} = \frac{\mathbf{E}_{i(k)}(\mathbf{A}_{(k)}) - \mathbf{E}_{i}(\mathbf{A}_{1(k)}, \cdots \mathbf{A}_{j(k-1)} \cdots, \mathbf{A}_{N(k)}))}{\mathbf{A}_{j(k)} - \mathbf{A}_{j(k-1)}}. \tag{2.78}$$

Rearranging equation (2.77)

$$\underline{E}_{(k)} = -[\underline{J}_{(k)}]\underline{D}_{(k)} \tag{2.79}$$

where

$$\underline{D}_{(k)} = \underline{A}_{(k+1)} - \underline{A}_{(k)}. \tag{2.80}$$

Equation (2.79) can now be solved for $\underline{D}_{(k)}$ using a standard Gauss elimination procedure. The corrected or improved vector is given by $\underline{A}_{(k)} + \underline{D}_{(k)}$ where $\underline{A}_{(k)}$ is the initial vector. A flow diagram of this numerical procedure is illustrated in Figure 2.9.

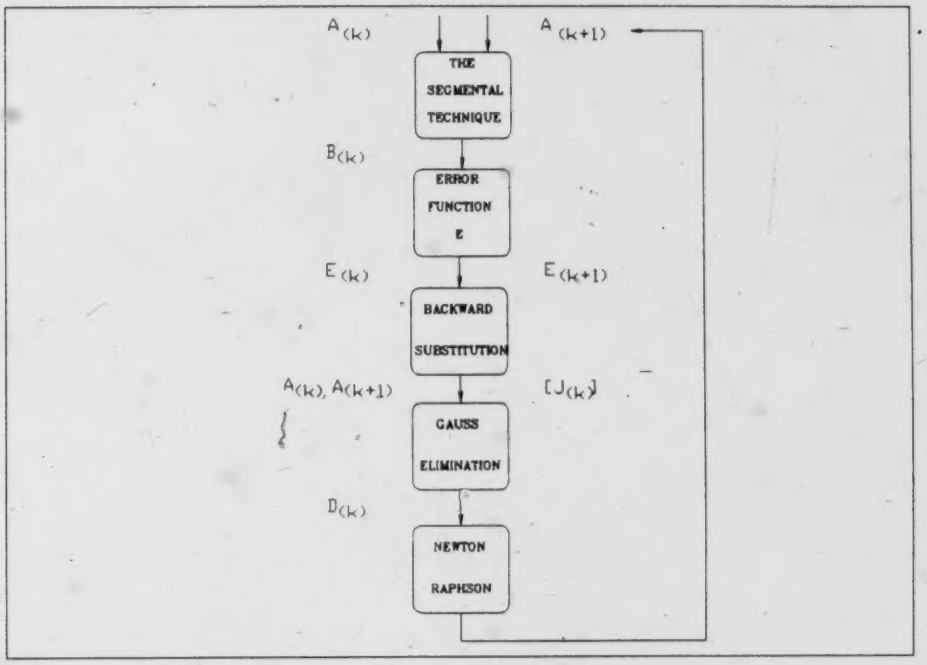


Figure 2.9 Flow Diagram of Numerical Shooting Technique

It is important to understand that finding roots of nonlinear equations or systems of nonlinear equations is a very difficult task in general. Root finding techniques such as the Newton Raphson false position algorithm described above only work well when good initial estimates of the solution are used or when the function is "*well behaved*". How close the initial estimate must be to the correct solution depends on the complexity of the functions involved in a given problem. For example, as will be discussed in detail later, many problems require initial estimates of the solution to be very close to the correct solution in order for the above false position method to converge to the solution desired.

CHAPTER
THREE

VERIFICATION

In Chapter 2, the segmental solution technique for analyzing large deflections of a rod was introduced. In order to verify the accuracy of this method, rod problems with point loads, uniform horizontal and vertical loads w_x, w_y as well as uniform normal and tangential loads w_n, w_t are considered in this chapter. The segmental technique was previously developed for analyzing large deflections of a rod with only one equilibrium shape. However, the problems being considered here are ones which admit multiple solutions. Uniqueness is not guaranteed and multiple equilibrium shapes are possible for a given load, (see for example Navaee [22]). Consequently, it is necessary to confirm the presence of multiple equilibrium configurations of a rod under various loading conditions. In the following sections, multiple solutions of a cantilever obtained using the segmental solution technique are compared with their analytical counterparts for a single concentrated end load P, a uniform distributed vertical load w_y and a uniform distributed normal load w_n. Loading conditions of the cantilever are shown in Figure 3.1. The origin of the cantilever for a particular loading is chosen so as to simplify the problem. The reason for doing so will become apparent in later sections. Also, a semi-circular arch subject to a horizontal concentrated load P at mid-span is investigated and compared with an analytical solution. In all cases it was found that the segmental solution technique gives an accurate description of the nonlinear deformation of the

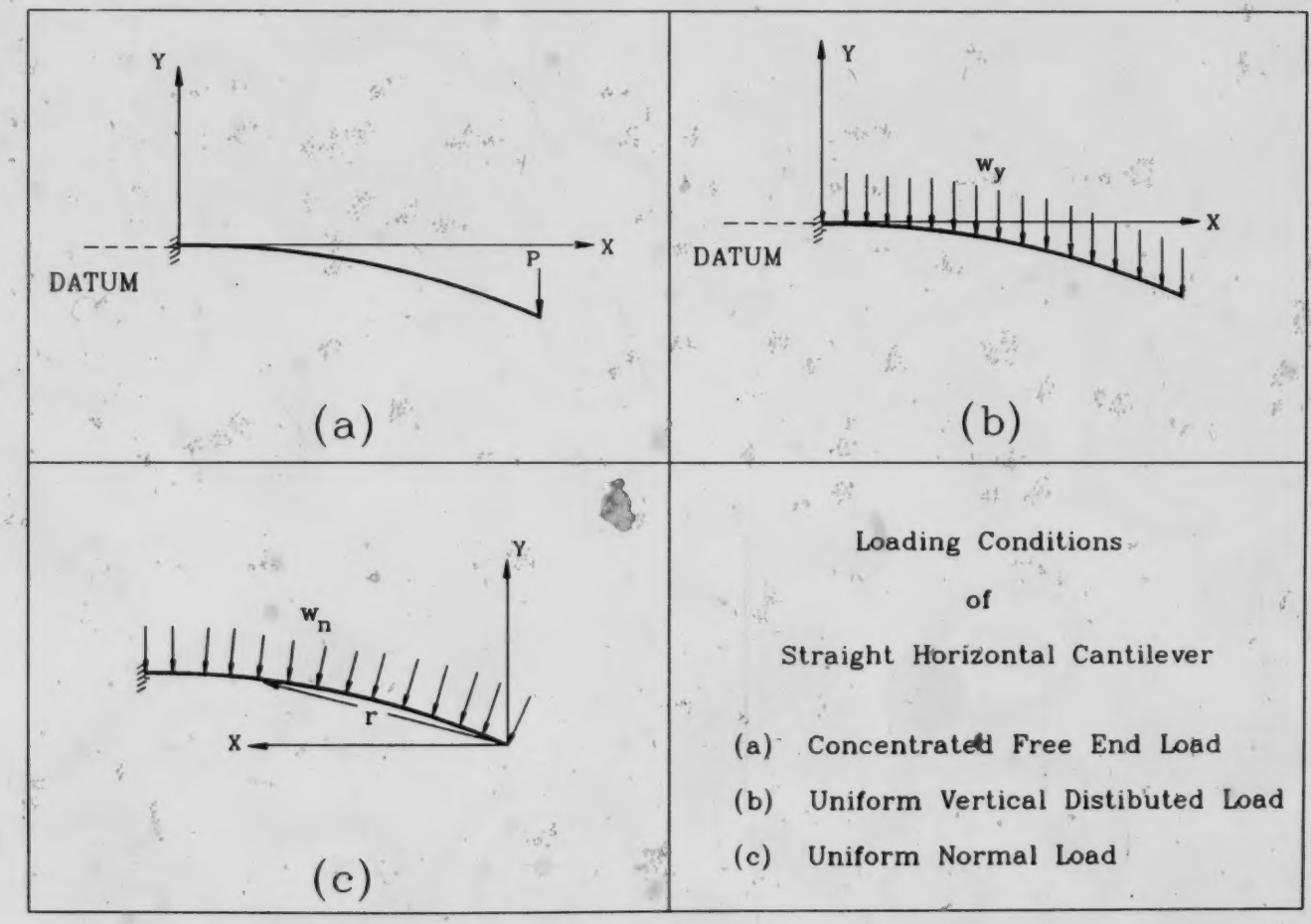


Figure 3.1 Cantilever Under Various Loading Conditions

cantilever, as long as a sufficient number of segments is used. The number of segments required to provide an accurate description of the problem depends on the amount of deformation involved and is discussed in later sections. In addition, the segmental solution technique is able to identify multiple equilibrium solutions when they exist.

3.1 Point Load P

Frisch-Fay [4] has derived an analytical solution for the large deformations of a horizontal cantilever under a vertical point load at the free end as shown in Figure 3.1(a). The details of this solution follow. Let ψ be the angle of the tangent of the cantilever with respect to the X-axis a distance s away from the fixed end and ψ_o be the corresponding angle at the free end. Introducing the variable ϕ, and the parameter p, which satisfy the following equations

$$1 + \sin\psi = (1 + \sin\psi_0)\sin^2\phi, \tag{3.1}$$

and

$$p^2 = \frac{(1 + \sin\psi_o)}{2}, \tag{3.2}$$

the solution to this problem can then be written in terms of standard elliptic integral as

$$q = \int_{\phi_1}^{\frac{\pi}{2}} \frac{d\phi}{(1 - p^2\sin^2\phi)^{\frac{1}{2}}} = K(p) - F(p, \sin^{-1}\frac{1}{p\sqrt{2}}) \tag{3.3}$$

and

$$s = \frac{[F(p,\phi) - F(p,\phi_1)]}{k}, \tag{3.4}$$

$$x = h(\cos\phi_1 - \cos\phi), \tag{3.5}$$

$$y = \frac{[2E(p,\phi_1) - F(p,\phi_1) + F(p,\phi) - 2E(p,\phi)]}{k}, \tag{3.6}$$

where

$$\phi_1 = \sin^{-1}(\frac{1}{p\sqrt{2}}), \tag{3.7}$$

$$k = (\frac{P}{EI})^{\frac{1}{2}}, \tag{3.8}$$

and

$$q^2 = \frac{PL^2}{EI}. \qquad (3.9)$$

Here x and y are respectively the horizontal and vertical displacement of the cantilever at a distance s away from the fixed end. The non-dimensionalized applied concentrated load is given by q in equation (3.9). K(p), E(p,ϕ) and F(p,ϕ) are respectively the complete elliptic integral, the elliptic integral of the first kind and second kind. From equation (3.3), for a given load q, the only unknown is the modulus p. A bisection root finding method with a lower bound of $p_{min}=(1/2)^5$ and upper bound of $p_{max}=.9999999999999999$ is employed to numerically compute p. A value of $p < (1/2)^5$ results in equation (3.7) being undefined and $p \geq 1$ leads to K(p) unbounded in equation (3.3). Physically, a modulus of $p = 1$ implies that the free end tangent with respect to the x-axis $\psi_o=90°$ indicating that the slope at the free end is vertical. Using p found from equation (3.3), equation (3.4) is used to solve for the slope ϕ_1, for any position s along the rod. Also, equations (3.5) and (3.6) respectively yield the vertical and horizontal coordinates of the cantilever at any location s.

Consider, as an example, the horizontal cantilever shown in Figure 3.1(a) with the following properties,

$$L = 1.0\ m, \qquad EI = 10.78125\ N.m^2,$$

$$1/R = 0\ m^{-1}, \qquad q = (PL^2/EI)^{.5} = 5.$$

Using the bisection method, the modulus p was found to be $p = 0.9999376728424472$. The horizontal and vertical coordinates at the free end of the rod evaluated using

equations (3.5) and (3.6) are tabulated in Table 3.1. Also shown is the slope ψ_o at the free end of the beam given by the appropriate rearrangement of equation (3.1). In all cases, ψ_o is equal to the notation γ_2 used in the derivation of the segmental approach.

Table 3.2 shows the numerical results obtained using the segmental technique for different numbers of segments. The numerical results approach the analytical solutions monotonically as the number of segments increase. The segmental technique with 2000 segments differs the analytical results by a maximum of 0.16%. Even though 2000 segments are used, numerical efficiency of this procedure is illustrated by the fact that the whole numerical iterative process is completed using a PC type computer in only a few minutes. Given a reasonable estimation of the initial unknowns, only a few iterations are required to converge to the desired solutions. Therefore, it can be concluded that the segmental technique agrees satisfactorily with the analytical solution and is numerically efficient in solving problems of this kind.

Table 3.1 Analytical Results of Cantilever Subject to a Free End Point Load P (Geometry of Free End)

X/L	Y/L	γ_2
0.2828074538111195	-0.882730526182732	-1.54846647144117

Table 3.2 Numerical Results of Cantilever Subject to a Free End Point Load P (Geometry of Free End)

# Segments	X/L	Y/L	γ_2
75	0.2696513066468	-0.8857840129169	-1.562961266457
123	0.2750562189956	-0.8845511770331	-1.556990954204
250	0.2789833514671	-0.8836363311167	-1.552666633891
500	0.2809079763399	-0.8831822670900	-1.550551443571
750	0.2815438996930	-0.8830314258671	-1.549853135693
1000	0.2818608204750	-0.8829561027265	-1.549505231364
2000	0.2823349087472	-0.8828432405612	-1.548984926246

Frisch-Fay's [4] derivations consider only one solution. In order to locate all possible equilibrium configurations, a plot is made of the non-dimensional moment ($\Lambda = ML/EI$) computed by the segmental technique at the free end of the cantilever (Λ_2) against the dimensionless moment initially guessed at the start of the cantilever (Λ_1) as shown in Figure 3.2 for a dimensionless load $q = 5.0$. Since the moment at a free end of the cantilever is identically zero, the three equilibrium configurations correspond to points where the ordinate Λ_2 is zero. It is essential to note that this function is quite complex and has a very steep gradient near $\Lambda_1 = -7.08$. This behaviour makes it imperative that the initial estimate of this root be very good if the false position algorithm described in the previous chapter is to be successful. From Figure 3.2, the three Λ_2 intercepts correspond to $\Lambda_1 = -7.08$, -4.32 and $+7.07$ indicate three possible equilibrium configurations within the range of Λ_1 used. If one considers the physical arrangement of the cantilever, the maximum moment at the fixed end due to the applied

load P at a lever arm of length L must be less than or equal to PL. To see if any more cases where $\Lambda_2 = 0$ exist, the segmental method was used with values of Λ_1 from $-PL^2/EI$ to PL^2/EI. No additional Λ_2 intercepts were found. It can thus be concluded that for $q = 5.0$ only three equilibrium configurations are possible and these are shown in Figure 3.3.

Also shown in Figure 3.3 are the shapes found by Navaee [22]. There is no apparent difference between his shapes and those found from the segmental solution technique. Note that the shapes obtained by Navaee [22] were computed via elliptic integral similar to the solution in equations (3.1) - (3.9). The exact number of multiple equilibrium solutions depends on the load parameter q. For $q < 3.214$ only one equilibrium solution occurs while if $q = 3.214$ two equilibrium solutions are present. If $3.214 < q < 7.142$ then there are three solutions and further increases in q result in more equilibrium solutions. One of the ways to determine the exact number of solutions for a given loading is to inspect a plot of fixed end moment versus free end moment and look for the zero free end moment conditions as illustrated in Figure 3.2.

Navaee [22] examined this problem of multiple solutions using a few different methods. Some of these methods involve evaluating the closed form analytical solution using elliptic integral. However, since it is often not possible to compute the exact solutions to a problem, the segmental approach is a feasible and efficient option.

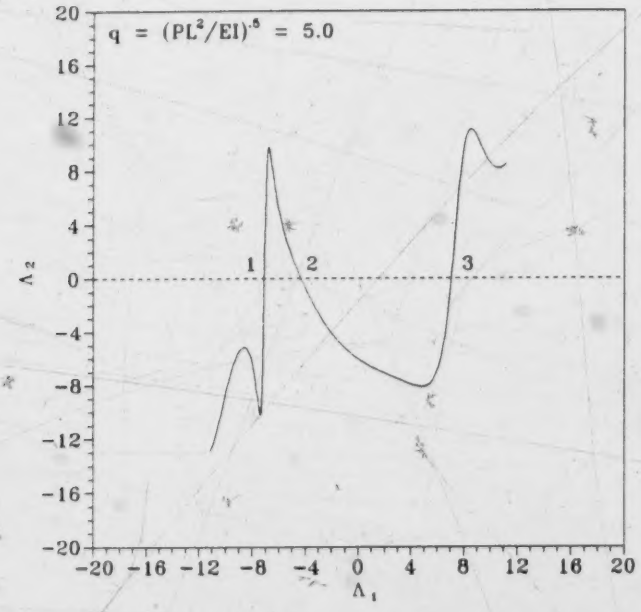


Figure 3.2 Free Node Moment Λ_2 vs Fixed Node Moment Λ_1 of the Cantilever Under Concentrated End Load

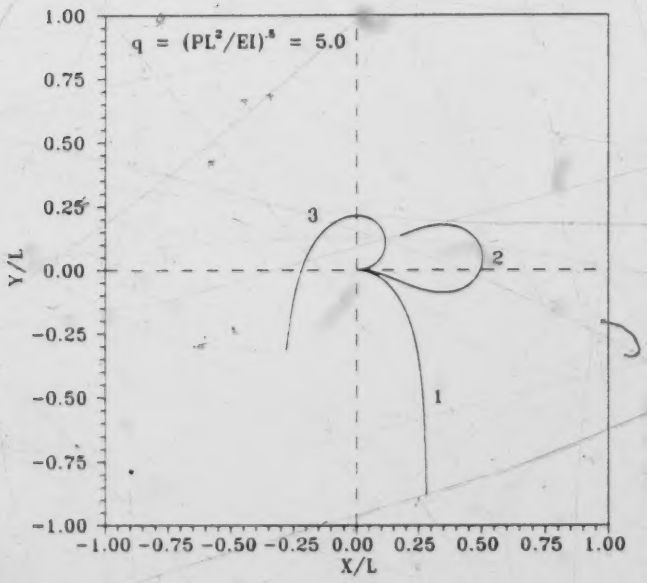


Figure 3.3 Multiple Equilibrium Solutions of a Cantilever Under Free End Load

3.2 Uniform Distributed Vertical Load w_y

Navaee [22] also compiled solutions of a cantilever beam under uniform distributed vertical loading w_y as shown in Figure 3.1(b) and validated them experimentally. Figure 3.4 shows the end moment vs start moment plot similar to the previous case for a non-dimensional load parameter $q_y = (w_y L^3/EI)^{.5} = 8.5$. Three separate equilibrium solutions are evident here because of the three zeros for Λ_2 which are labelled 1, 2, 3. Again the exact number of equilibrium configurations depends on the load parameter q_y.

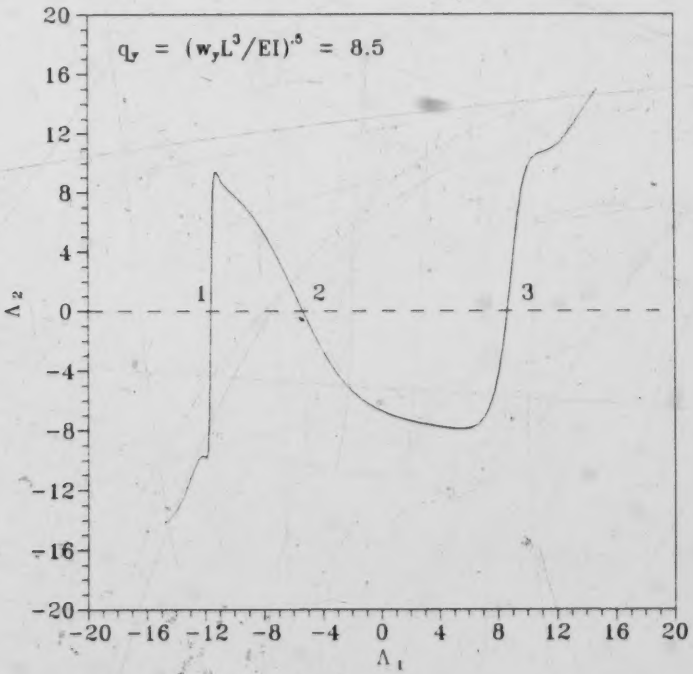


Figure 3.4 Free End Moment Λ_2 vs Fixed End Moment Λ_1 of the Cantilever Under Vertical Distributed Load

Figure 3.5 shows the resulting three equilibrium configurations computed by the segmental solution method. Also shown are the shapes computed numerically by Navaee

[22]. Again there is no observable difference between these solutions.

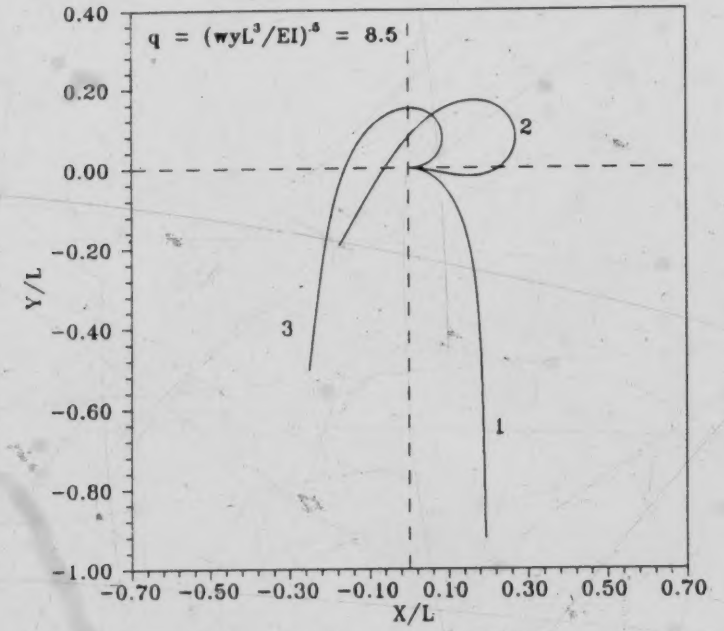


Figure 3.5 Multiple Equilibrium Solutions of Cantilever Under Vertical Distributed Load

3.3 Normal Load w_n

Mitchell [8] derived the analytical solution of a cantilever subject to a uniform normal load $q_n = (w_n L^3/EI)^{.5}$ as shown in Figure 3.1(c). The deformed shape of the cantilever is given by

$$3\theta - \sin^{-1}(q_n \rho^3) - 4\sin^{-1}(q_n \rho_L^3) + 3\beta \tag{3.10}$$

and

$$F(\Omega,k) = 2q_n^{1/2}(3)^{\frac{1}{4}} \tag{3.11}$$

where

$$\Omega = \cos^{-1}\left(\frac{1-(\sqrt{3}+1)q_n^{2/3}\rho_L^2}{1+(\sqrt{3}-1)q_n^{2/3}\rho_L^2}\right) \tag{3.12}$$

and

$$k^2 = \frac{2-\sqrt{3}}{4} \tag{3.14}$$

$$\rho = \frac{r}{L}, \tag{3.13}$$

where $F(\Omega,k)$ is the elliptic integral of the first kind. The polar coordinate system as well as the physical representation of β are shown in Figure 3.6. In this case, β is the angle at location r and θ is the angle at the free end of the cantilever. ρ_L in equation (3.12) is the dimensionless form of r at the clamped end of the cantilever.

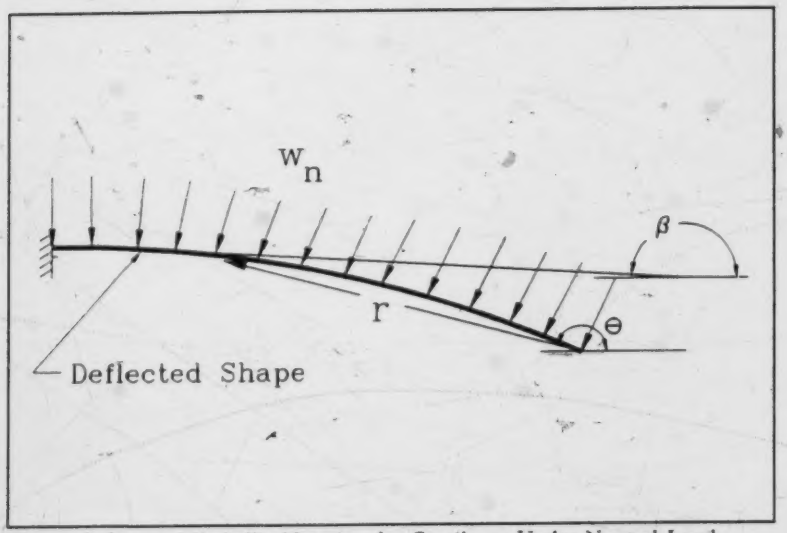


Figure 3.6 Mitchell's Notation for Cantilever Under Normal Load

Table 3.3 compares the results obtained using the 1000 segments with the segmental approach and the analytical solution. Instead of defining the origin at the fixed end of the cantilever, the origin is selected at the free end of the beam which simplifies the problem for the segmental solution. That is, the origin is defined at the free end of the cantilever as indicated in Figure 3.1(c). Since the deformed shape of the beam is not known before hand, the moment, tension and shear at the fixed end are unknowns. However, moving the origin to the free end means that only the free end slope γ_1 is unknown with the free end tension, shear and moment being identically zero. With fewer initial unknowns, the problem is thus simplified. Since the slope at the fixed end is zero, the equilibrium solutions correspond to points where $\gamma_2 = 0$. If one used the same approach as for the cases with cantilevers, that is prescribing different values of the kinematic variable γ_1 at the free end and computing the corresponding slope γ_2 at the fixed end without changes in tension, shear and moment this would essentially invoke a

rigid body rotation of the cantilever by an angle of γ_1. This would not provide any information pertaining to the multiplicity of the solutions to the problem. To overcome this difficult, the problem is posed with a couple M_1 applied at the free end. For various values of initial slope γ_1, the shooting technique is used to find the free end moment M_1 (Λ_1 in non-dimensional form) which ensures $\gamma_2 = 0$. A plot of Free End Moment Λ_1 vs Free End Slope γ_1 with the condition $\gamma_2 = 0$ for different values of q_n is shown in Figure 3.7. Only one equilibrium solution (Λ_1-intercept) for each load was found for the range considered. The prescribed free end rotation γ_1 has been computed for the range $0 < \gamma_1 < 4\pi$. As the prescribed rotation γ_1 is increased, the value of free end moment Λ_1 continues to decrease. This presents strong evidence that the cantilever loaded under a normal load has only one equilibrium solution regardless of the magnitude of the applied load.

Table 3.3 Analytical and Numerical Results of a Cantilever Subject to Uniform Normal Load (Geometry of Free-End)

q_n	ANALYTICAL SOLUTION		SEGMENTAL TECHNIQUE	
	X/L	Y/L	X/L	Y/L
1.8161	0.922005	-0.35395	0.922554	-0.35098
3.7882	-0.00206	-0.82349	0.000534	-0.82256
5	-0.40269	-0.47883	-0.38827	-0.48909
10	-0.04690	-0.00006	-0.04531	-0.00548
15	0.194960	-0.04521	0.194110	-0.05414

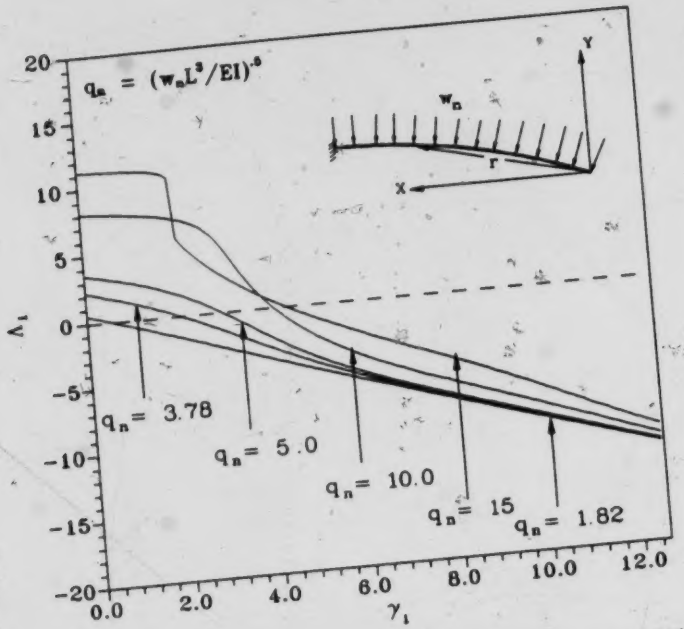


Figure 3.7 Fixed Node Moment A_1 vs Fixed Node Slope γ_1 of the Cantilever Under Normal Load

The corresponding deflected geometries of the cantilever are shown in Figure 3.8. roken lines denote Mitchell's analytical solution whereas the solid lines are uted using the segmental approach. It is observed that at higher loads such as $q \geq$ e segmental approach with 1000 segments deviates noticeably from the computed lytical solution. This is due to the breakdown of the small angle approximation used leriving this technique. Large deformations, indicating large relative rotation between ends, necessitates the need to increase the number of segments hence reducing the ative rotations between successive segments.

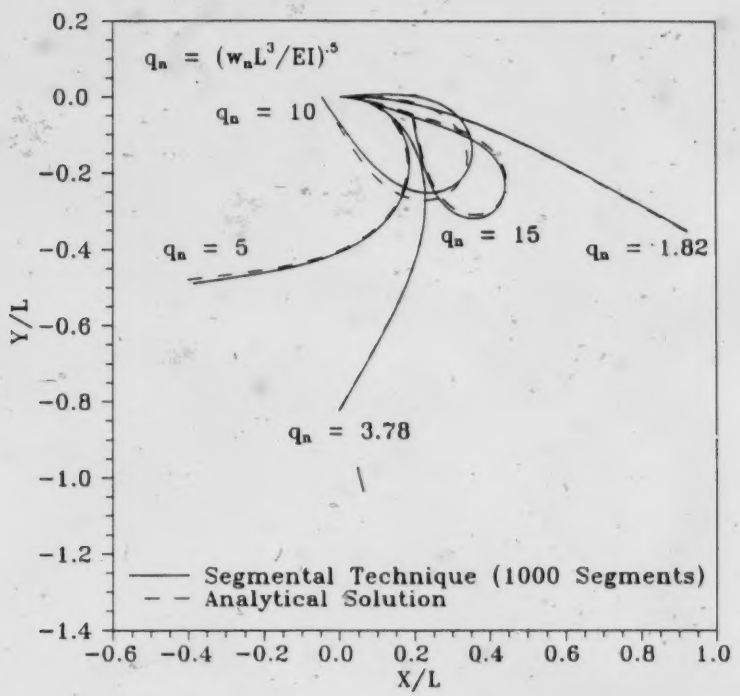


Figure 3.8 Analytical and Numerical Results of the Cantilever Under Normal Load

Table 3.4 Analytical and Numerical Results of a Cantilever Subject to Uniform Normal Load q_n = 10 and 15 (Geometry of Free End)

q_n	ANALYTICAL SOLUTION		SEGMENTAL TECHNIQUE		# Segments
	X/L	Y/L	X/L	Y/L	
10	-0.04690	-0.00006	-0.04604	-0.00079	2000
15	0.194960	-0.04521	0.195815	-0.04465	2000

Figure 3.9 represents the deflected geometry of the cantilever with 2000 segments for q_n = 10 and 15 respectively. The deformed shapes are now in better agreement with the analytical solution as shown graphically and as listed in Table 3.4.

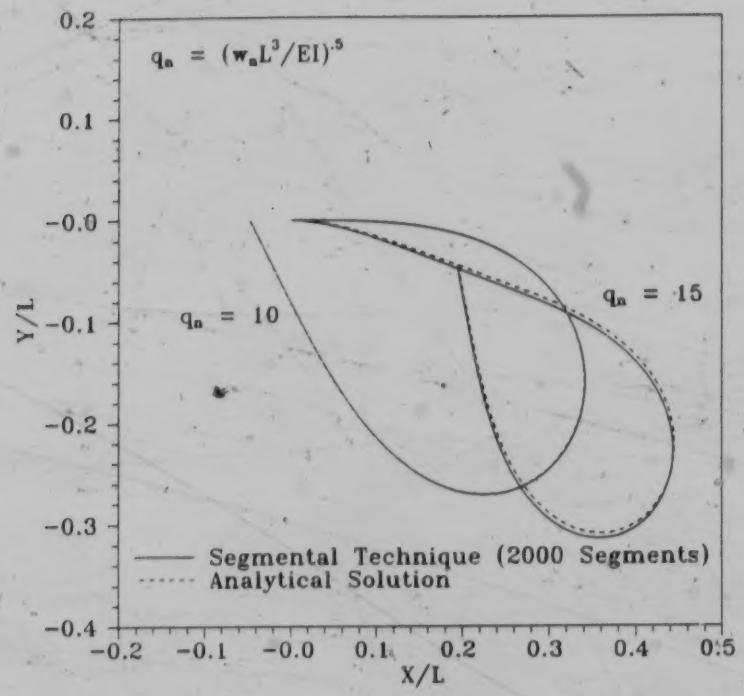


Figure 3.9 Analytical and Numerical Results of the Cantilever Under Normal Load

The minimum number of segments required to produce acceptable answers is dependent on the amount of deformation which directly relates to the loading conditions. For an applied load parameter $q_n = 10$, the segmental solution agrees satisfactorily with its analytical counterpart only when more than 2000 segments are used. The deflected shape of the cantilever undergoes 239.7° rotation. Therefore, as an overall average, 8 or 9 segments per degree change of rotation is recommended to achieve acceptable results.

3.4 Pin End Deep Arch

Large deformations of rods with only one initial unknown have been examined so far. However, there are numerous problems where more than one initial value is not known. Consider a semi-circular arch pinned at both ends as shown in Figure 3.10. The arch is subject to a horizontal point load P applied at the crown.

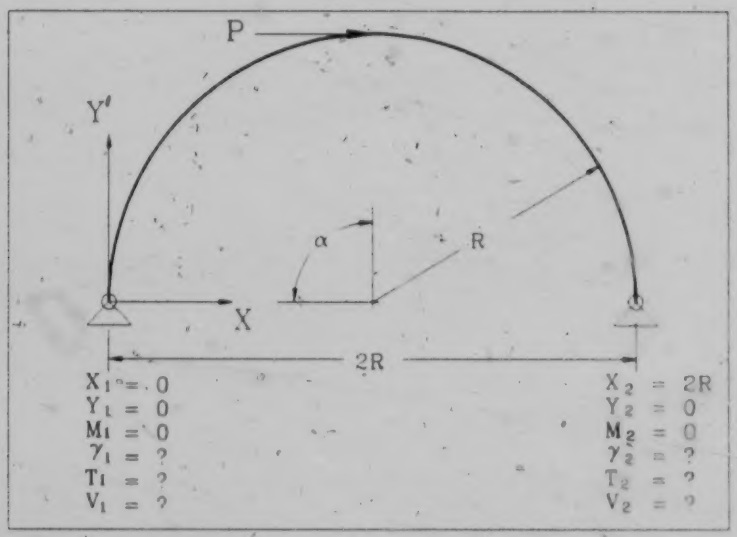


Figure 3.10 Boundary Conditions of the Semi-Circular Arch

With the origin of the coordinate system at the pin on the left, the initial knowns are $X_1 = 0$, $Y_1 = 0$, $M_1 = 0$ and the corresponding unknowns are T_1, V_1 and γ_1. In this case, three end or known boundary conditions at the pin on the right have to be satisfied, namely $X_2 = 2R$, $Y_2 = 0$ and $M_2 = 0$. Specifically the deformation of deep arches is an often studied problem. DaDeppo and Schmidt [19] investigated the sideways deflections of deep circular arches under concentrated loads.

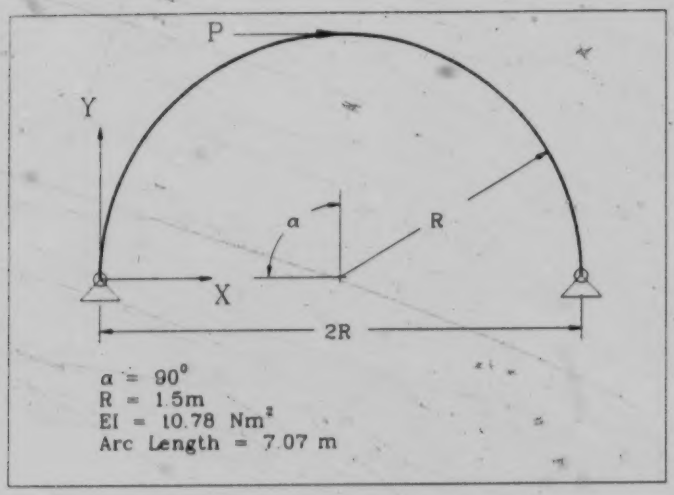


Figure 3.11 Pin End Semi-Circular Arch Subject to Horizontal Mid-Span Point Load

For example is a semi-circular arch with properties as shown in Figure 3.11, pinned at both ends and loaded at the crown. DaDeppo and Schmidt [19] employed a nonlinear analysis based on the classical Bernoulli-Euler theory of the elastica. The solution to the problem is expressed in terms of elliptical integrals of the first and second kind, however the solution involves the simultaneous satisfaction of equations for the left and right sides of the arch. A system of ten equations, five of which are transcendental, must be solved simultaneously. DaDeppo and Schmidt [19] reported that solving a system of transcendental equations was a difficult task and a digital computer was used to perform the numerical computation. Numerically examining this arch using the segmental approach requires the technique to accommodate an externally applied point load between segments. Figure 3.12 depicts the free and deformed shape of the arch under a horizontal concentrated applied load at its crown. The corresponding deformed

configuration is essentially the same as with DaDeppo's analytical solutions. However, the fact that there may be multiple equilibrium solutions was not discussed by these authors. Those found using the segmental technique are discussed in Chapter Four.

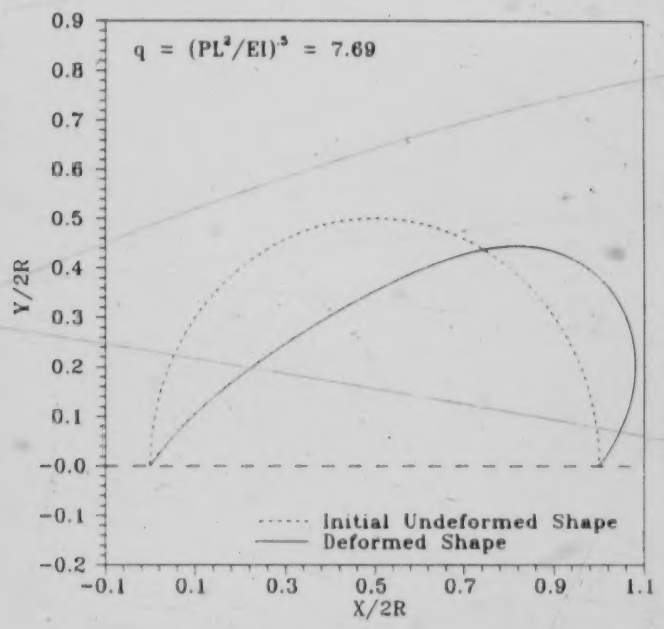


Figure 3.12 Free and Deformed Shape of the Arch Under Horizontal Crown Load

The segmental approach to the above problem not only yields very accurate results for large deflections of rods but can also be used to identify multiple solutions if they are present. This is very straightforward when there is only one initial unknown. However, finding multiple solutions for problem that have several unknowns require special modifications of the shooting procedure. These are discussed in the following chapters.

CHAPTER
FOUR

RESULTS AND DISCUSSIONS

In Chapter Three, the segmental approach was used to examine the multiplicity of solutions for a cantilever subject to various loading conditions. Also, it was suggested that structures such as deep arches exhibit multiple equilibrium solutions for various loading conditions. Even though some of the computed deformed configurations were verified using analytical solutions, many of them are "*counter-intuitive*" prompting questions of their stability.

This chapter begins by introducing the notion of stability using the minimum potential energy theorem. This notion of stability will be used as a basis for a limited evaluation of the stability of the computed deformed configurations. In addition to evaluating the stability of the solutions, the deformed configurations not verified by the analytical solutions were checked by ensuring overall static equilibrium. The investigations of cantilevers under different loadings are again discussed using this restricted concept of stability. Arches with different support and loading conditions are also considered to illustrate the realization of multiple solutions using the segmental approach. In all cases, the segmental technique has been able to identify multiple solutions if present. The evaluation of the stability of the solution using the potential energy criterion is used to provide indications of the relative stability of the multiple solutions.

4.1 Minimum Potential Energy Criterion for Stability

While a complete and precise definition of stability is given using the Lyapounov definition [34], the more widely used energy criterion is utilized in the following. It is realized that only in certain circumstances is the energy criterion a necessary condition for Lyapounov stability. An expression of the energy criterion for stability is given by Knops and Wilkes [34] as:

> *"For an elastic body under conservative loads and subject to isothermal conditions, a necessary and sufficient condition for stability of an equilibrium position is that the potential energy of the system assumes at the equilibrium position a weak relative minimum in the class of virtual displacements satisfying the kinematic constraints."*

From this definition it is further interpreted that

(i) Since the precise definition of virtual is not stated and can be a very ambiguous concept, all kinematically admissible displacement fields should be considered. These are not necessarily infinitesimal.

(ii) As not all possible kinematically admissible fields are considered, the above condition is treated only as a necessary one.

(iii) The idea of a weak relative minimum will be imprecisely interpreted as being a "local" minimum in some sense which will be described.

(iv) The kinematically admissible displacement fields will be found from a static analysis. This is all that can be provided by the

segmental technique outlined above.

(v) For the planar rods considered, the potential energy is given by

$$\hat{E} = W - L \tag{4.1}$$

where

$$W = \int_0^s \frac{1}{2}\frac{M^2}{EI} ds, \tag{4.2}$$

$$L = \int_0^s \rho_0 \underline{b} \cdot \underline{\chi}(s) ds + \sum_{i=1}^{n} \underline{F}_i \cdot \underline{\chi}_i(s_i). \tag{4.3}$$

Here $\hat{E}$ is the potential energy of a conservatively loaded elastic rod, W is the corresponding strain energy and L is the load potential due to the applied forces $\underline{F}_i$ and body force $\underline{b}$ through displacement $\underline{\chi}$ at position s.

In the present investigation, the potential energy is calculated for various configurations of the rods, by considering, in some consistent manner, other equilibrium conditions which satisfy the kinematic constraints on the boundaries. This is done by "*perturbing*" the initial conditions of the known solution. These perturbed conditions can result in displacement fields which are different from the actual solution. Even though the perturbed conditions will not generate **all** kinematically admissible displacement fields, it does give at least a partial understanding of the stability of these solutions.

4.2 Energy Considerations of Cantilevers

In Chapter Three several solutions for the cantilever under various loads were verified by comparison with the corresponding analytical solutions. In addition to the possibility of these solutions, the stability of conservatively loaded cantilevers can be assessed using the Criterion of Minimum Potential Energy. The potential energy defined in equation (4.1) - (4.3) is evaluated using the segmental approach for the cantilever under concentrated and distributed vertical load. The computed potential energy values will form the basis for evaluating the stability of the computed solutions.

4.2.1 Cantilever Under Free End Concentrated Load

The horizontal cantilever under a vertical concentrated non-dimensional load $q = 5.0$ was shown in Figure 3.1(a) and is reproduced in Figure 4.1 where the values of non-dimensional clamped end moments Λ_1 are included.

In the solution technique described, free end moment Λ_2 was found as a function of the fixed end moments Λ_1. Solutions to the actual boundary value problem were determined when the condition of zero free end moment, $\Lambda_2 = 0$, was established. Since the loading is conservative, the potential energy $\hat{E}$ of the cantilever can also be evaluated. Instead of plotting Λ_2 vs Λ_1, the non-dimensional potential energy Π of the system can be plotted against Λ_1.

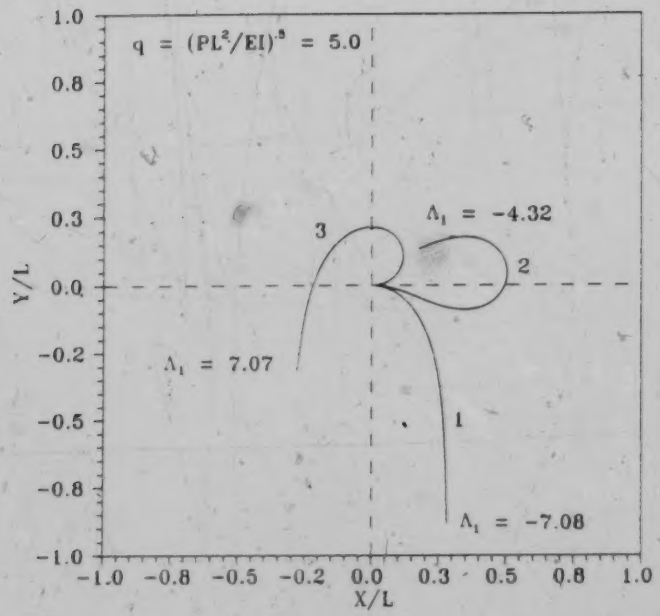


Figure 4.1 Multiple Solutions of a Cantilever Under Free-End Point Load

From equation (4.3), the load potential $\bar{L}$ is given by,

$$L = \underline{F}_i \cdot \underline{\chi}_i(s_i) - Fy(L) \qquad (4.4)$$

since the body force $\underline{b} = \underline{0}$ is considered. According to the energy criterion, an equilibrium configuration is stable if it corresponds to a local minimum in the potential energy. Such an "*energy plot*" is shown (the Λ_2 vs Λ_1 plot of Figure 4.2 is also included) Figure 4.2 using dimensionless parameters Λ_1, Λ_2 and Π where

$$\Pi = \frac{\hat{E}L}{EI} \qquad (4.5)$$

The three equilibrium shapes in Figure 4.1 correspond exactly to the three local minima of the energy curve and to the three zeros of the moment curve. As might be expected the energy corresponding to shape #1 is both a global and local minimum. Shape #3, while being a distinct local minimum, has a higher potential energy than Shape #1. The third shape, #2, has a somewhat indistinct local minima which can be interpreted to mean that it would be quite sensitive to disturbances (i.e., small changes in the boundary conditions or loading). Overall, this energy curve implies that shape #1 is globally the most stable while shape #3 appears locally stable. Shape #2, while being locally stable, appears very sensitive to the type of variations which have been considered in this analysis.

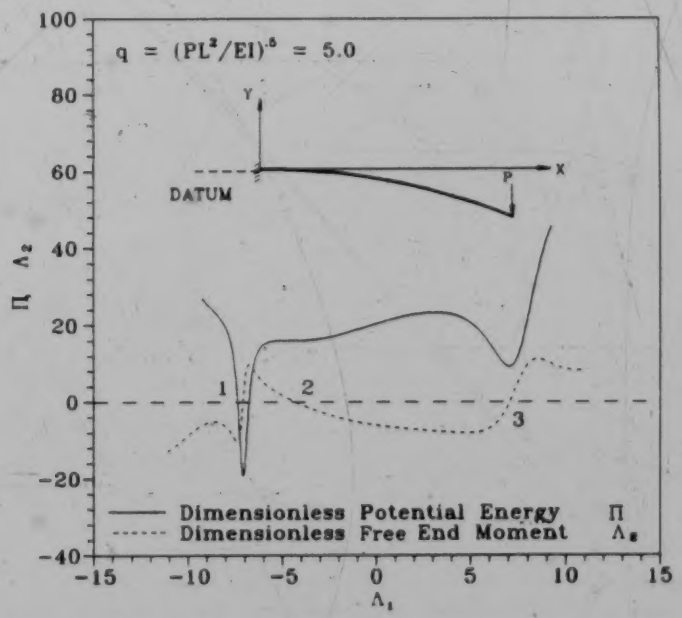


Figure 4.2 Combined Free End Moment Λ_2 and Energy Π vs Fixed End Moment Λ_1 For Cantilever Loaded at Free End

It must be remembered that the energy plot was determined by considering a particular class of kinematically admissible displacements. This class of kinematically admissible displacement fields involves only cases where the cantilever is in static equilibrium satisfying the kinematic boundary conditions. Here the kinematic boundary conditions are $X_1 = 0$, $Y_1 = 0$ and $\gamma_1 = 0$. Note that all the kinematic constraints are imposed at the start of the rod eliminating the necessity to satisfy kinematic constraints at the end of the rod. This will again be discussed below in conjunction with other problems in which kinematic boundary conditions exist at both boundaries. In this case, the energy plot was generated by considering different values of the fixed-end moment parameters Λ_1. By doing so, it is in effect "*perturbing*" the rod by prescribing end moments (i.e., $\Lambda_2 \neq 0$) which result in different displacement fields other than the solution of interest. This is, of course only a subset of all kinematically admissible displacement fields necessary to completely evaluate the energy criterion.

It is also of interest to note that Fried [21] considered the stability of a similarly loaded cantilever using a finite element computation. In his discussion, the eigenvalues of the tangent stiffness matrix were evaluated for each of the three configurations discussed above. If the eigenvalues were all positive, the configuration was pronounced stable and unstable if any one of them were negative. In his investigation, the cantilever is modelled using a total of seven quadratic elements each consisting of three nodal points. However, by evaluating the sign of the lowest eigenvalues, he judged that shapes #1 and #3 are stable while #2 is unstable. In spite of this result, Navaee [22] reports that configuration #2, while difficult to maintain, was possible to produce experimentally.

To detail the progression of the potential energy of the cantilever for ascending load parameters Figures 4.3 is included. As indicated in Chapter Three, three equilibrium solutions are found for $q > 3.71$ and only one when $q < 3.71$. Figure 4.3 shows only one local minimum for q ranging from 1.0 to 3.0. However, as the load increases the shape begins to form a very distinct notch. At $q = 4.0$ (Figure 4.3), three local minima in the potential energy are found and these becomes more distinct as q is raised from 4.0 to 5.0. This shape remains qualitatively the same until $q = 7.45$ after which two additional local minima are found. These additional minimum values correspond identically to the next two equilibrium shapes described by Navaee [22].

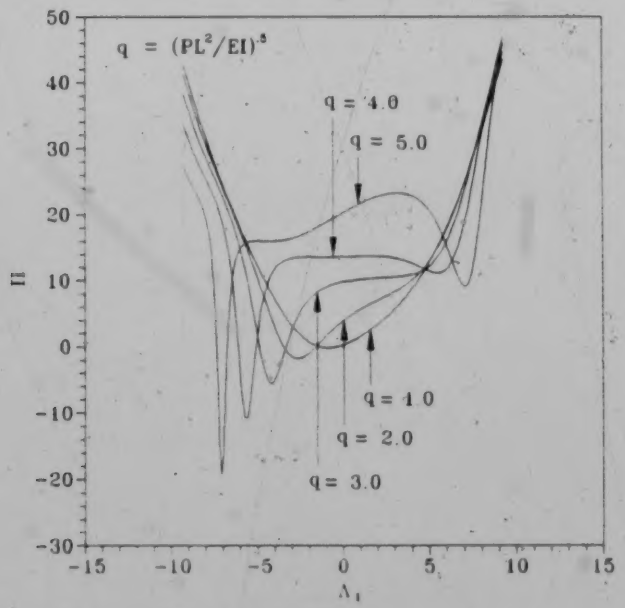


Figure 4.3 Potential Energy For Concentrated Loads

4.2.2 Cantilever Under Distributed Load

A similar analysis can be done for the horizontal cantilever under uniform vertical distributed loading (Figure 3.1(b) and Figure 3.5). Again three deformed configurations, labelled 1, 2 and 3, are found at $q_y = 8.5$ (see Figure 4.4). The corresponding energy plot (Figure 4.5) shows a form qualitatively like that of the concentrated load case (Figure 4.2) with similar interpretations. Again superimposed on this curve is the fixed end moment Λ_1 versus free-end moment Λ_2 curve showing the three solutions. It is also interesting to note that Navaee [22], who reported experimental results for this case, comments that shape #2 was difficult to obtain and appeared to be "*neutrally stable*". This is interpreted to mean that this shape was very sensitive to perturbation and would easily move toward one of the other two equilibrium configurations. Again, the deformed configuration having the lowest minimum energy is shape #1 whereas shape #2 corresponds to the highest energy of the three equilibrium configurations.

The development of these local minima can again be seen by considering the energy plots as the distributed load parameter increases. Figure 4.6 shows the curves for $q_y = 2.0$, 4.0 and 8.5. The single equilibrium position for $q_y = 2.0$ corresponds to the single minimum of this curve. As the load increase, first to 4.0 and then to 8.5, the development of the three local minima are seen.

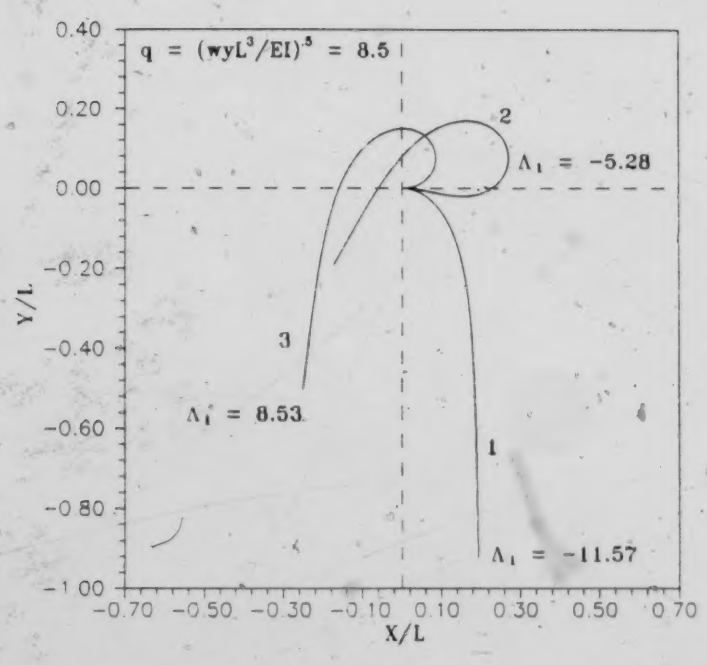


Figure 4.4 Multiple Solutions of the Cantilever Under Vertical Distributed Load

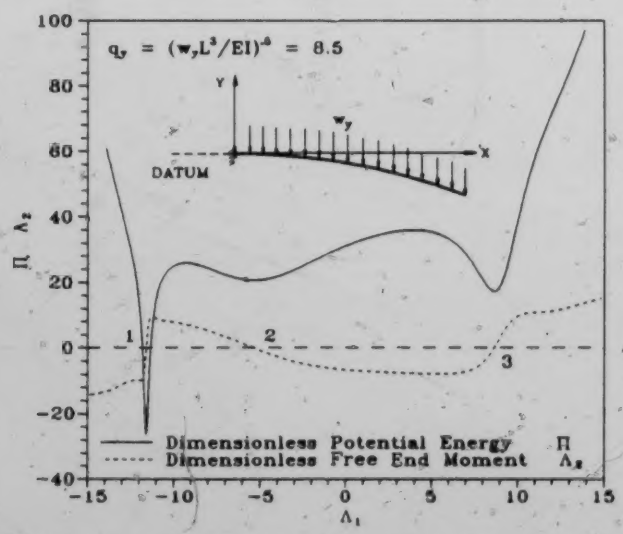


Figure 4.5 Combined Free End Moment Λ_2 and Energy Π vs Fixed End Moment Λ_1 of Cantilever Under Distributed Load

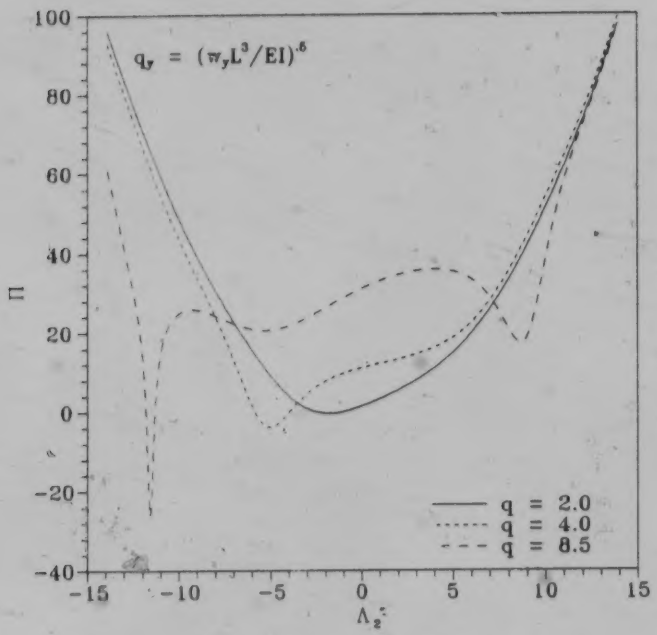


Figure 4.6 Potential Energy For Concentrated Loads

4.3 Arches

The analysis of the large deformation of arches is a popular class of structure to study because of their engineering significance. The following sections analyse arches with different support conditions using the segmental approach. The multiplicity of solutions and stability of the deformed configurations are examined. It is found that the method of solutions employed for analysing cantilevers can also be used when examining arches. The limitations of the segmental approach in realizing multiple solutions are illustrated through the formulation of different arch problems. The evaluation of the stability of the solutions using the minimum energy criterion analysis is also explored.

4.3.1 Zero Load Solution(s)

Consider a semi-circular arch ($\alpha = 90°$) pinned at one end, with a roller support at the other as shown in Figure 4.7.

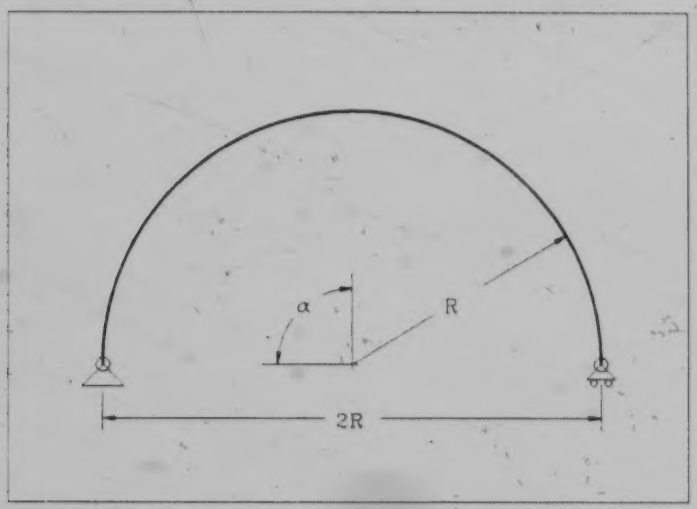


Figure 4.7 Semi-Circular Arch with Pin-Roller Supports

By inspection, the supports of the arch allow at least two possible equilibrium unloaded configurations (i.e., zero load solutions) as shown in Figure 4.8. Since some of the possible deformed configurations investigated later do not appear to be obvious, realizing that there are two zero load solutions assists in visualizing the development of certain shapes. The zero load solutions labelled #1 and #2 will be referred to in the following sections.

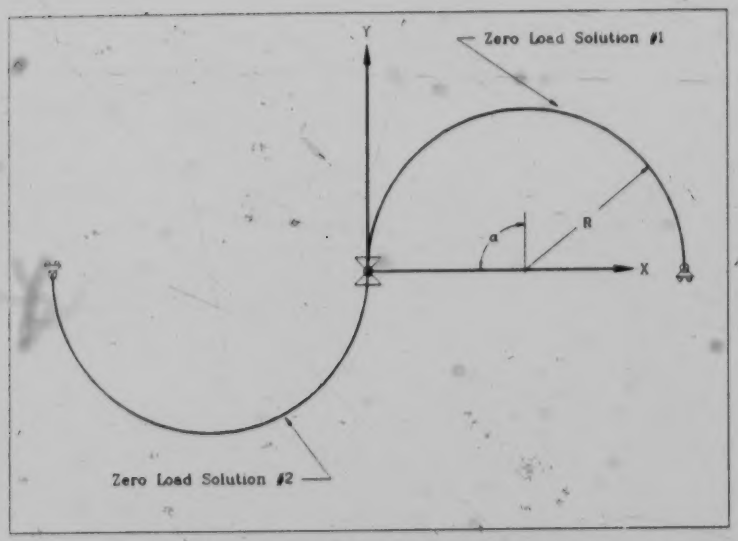


Figure 4.8 Possible Zero Load Solution(s) of Pin-Roller Supported Arch

Regardless of how the arch is loaded, this problem is somewhat different than those of the cantilevers above. This is due to the fact that there are kinematic constraints on both boundaries. While this fact is not overly important when considering the segmental technique to find the equilibrium configurations, it is important when developing a class of kinematically admissible displacement fields for the energy considerations. The initial values of the arch are $X_1 = 0$, $Y_1 = 0$, $M_1 = 0$ and $T_1 = V_1 = 0$ for the unloaded arch. If the arch is unloaded (i.e., undeformed) the initial value γ_1 is $\pi/2$ for zero load solution #1 and $3\pi/2$ for zero load solution #2. In order to consider the potential energy plot for the arch, the additional kinematic constraint, $Y_2 = 0$ must also be satisfied. This adds a complication since the segmental approach solves an initial value problem and the $Y_2 = 0$ condition would not normally be satisfied. Varying the kinematic variable γ_1 without satisfying the kinematic end constraint

$Y_2 = 0$ allows a rigid body rotation with no change in strain energy. For instance, by prescribing $\gamma_1 = \pi/4$, the computed deformed shape of the arch under no load condition is shown in Figure 4.9 with $Y_2 \neq 0$. The undeformed shape of the arch is also shown for comparison.

Figure 4.9 Rigid Body Rotation of Arch with no Change in Strain Energy

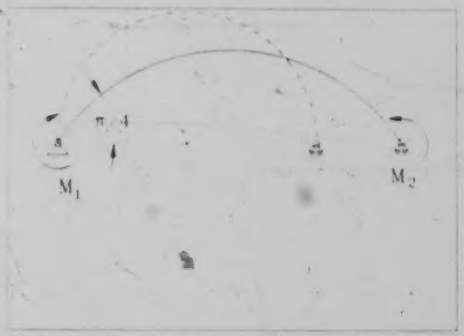


Figure 4.10 Positive Prescribed Moments at Both Ends Satisfying Boundary Condition $Y_2 = 0$

In order to impose the kinematic boundary condition $Y_2 = 0$, moments M_1 and M_2 (see Figure 4.10) are applied at the supports of the arch. The rationale for selecting applied moments instead of applying a tension or shear is simply because that the applied moments can easily be visualized as "*bending*" the arch to meet the desired end conditions. In this case, no vertical deflection of the roller support ($Y_2 = 0$) is the only required kinematic end condition. Even though the force boundary conditions $M_1 = M_2 = 0$ are not satisfied, the computed displacement field is kinematically admissible. While the segmental technique can solve the problem and satisfy all the boundary conditions, the solution for $M_1 \neq 0$, $M_2 \neq 0$ allows a class of kinematically admissible displacement fields to be generated and the potential energy calculated.

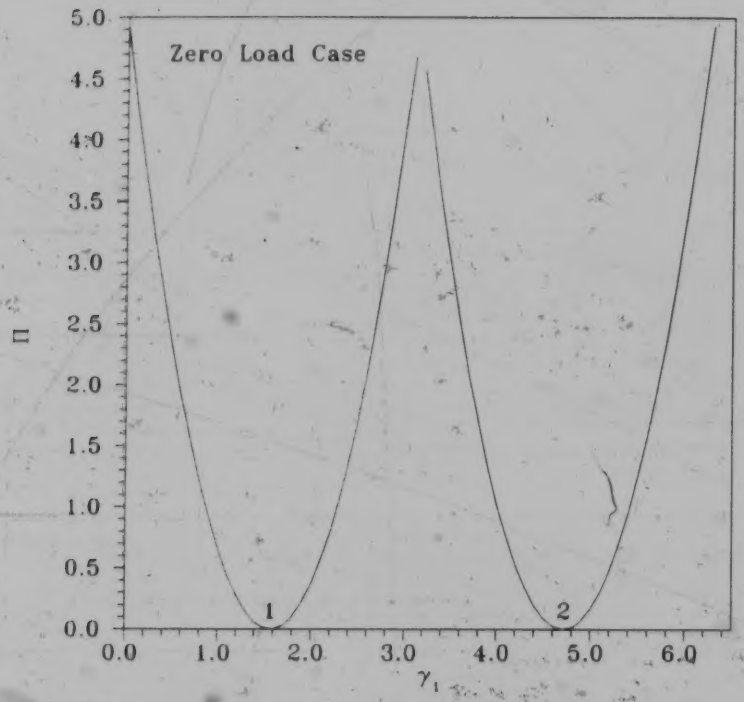


Figure 4.11 Potential Energy Π of an Unloaded Arch

The potential energy plot shown in Figure 4.11 is determined by selecting a series of initial angles, γ_1, and then using the segmental shooting technique to find M_1 and M_2 such that all kinematic constraints are satisfied. As expected, the potential energy of the deflected shapes show very distinct minima at $\gamma_1 = \pi/2$ and $\gamma_1 = 3\pi/2$. Any other prescribed γ_1 results in a higher potential energy which is the necessary stability condition for the minimum energy criterion. Moreover due to its periodic nature, the range of γ_1 considered "*sweeps*" through all possible initial slopes. Thus the two zero solutions in Figure 4.8 are the only possible shapes and are locally stable.

As described in Section 2.3, the segmental shooting technique requires six parameters at the start of a rod to initiate the procedure. As previously discussed these

six parameters are X_1, Y_1, γ_1, M_1, T_1 and V_1. Since only some of the initial values are known for a particular problem, a Newton Raphson false position routine is employed to correct or predict the initial unknown values. The number of initial unknowns is one of the factors that determine the complexity of the problem, especially when the determination of multiple solutions to the problem is required. This is illustrated in the following sections in which various arch problems having different numbers of unknowns are formulated.

4.3.2 Arch with Concentrated End Load

Consider the arch subject to a horizontal point load P applied at the roller support as illustrated in Figure 4.12. Static equilibrium on the arch requires that in the deformed configuration

$$+\rightarrow \sum F_x = 0 = -T_1\cos\gamma_1 - V_1\sin\gamma_1 + P, \tag{4.6}$$

$$+\uparrow \sum F_y = 0 = -T_1\sin\gamma_1 + V_1\cos\gamma_1 \tag{4.7}$$

Here these two equations involve three unknowns so that the initial tension T_1 and initial shear V_1 can be thought of as functions of the initial slope γ_1. As a result γ_1 is the only initial unknown. One kinematic constraint $Y_2 = 0$ and one force constraint $M_2 = 0$ are imposed at the end of the arch.

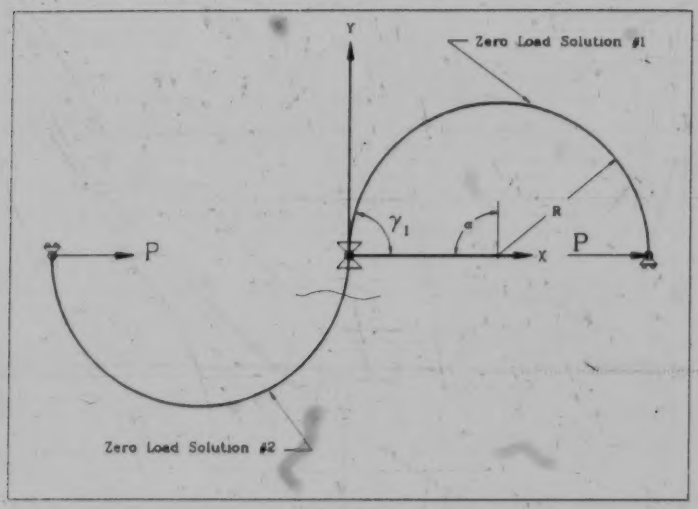


Figure 4.12 Semi-Circular Arch Subject to a Horizontal Point Load P Applied at The Roller Support

If the initial slope γ_1 is varied and plotted against either $Y_2 = 0$ or $\Lambda_2 = 0$, the result is shown in Figure 4.13 for a dimensionless load $q = 1.0$. The locations of $Y_2/2R = 0$ coincide with points of $\Lambda_2 = 0$. This implies that both kinematic and force constraints are satisfied at those points. The corresponding deformed configurations are shown in Figure 4.14 where the dash lines indicate the zero load solutions shown earlier. In this instance, the deformed configurations follow from the two unloaded shapes. Note that the slope γ_1 is varied form 0 to 2π meaning that all possible initial slopes, γ_1, of the arch have been considered and only two solutions are found for $q = 1.0$.

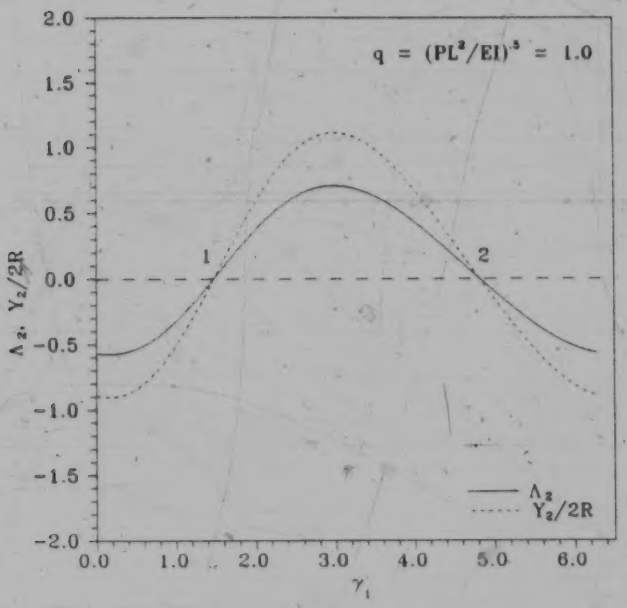


Figure 4.13 Combined End Moment Λ_2, $Y_2/2R$ vs Start Slope γ_1 of the Pin-Roller Support Arch

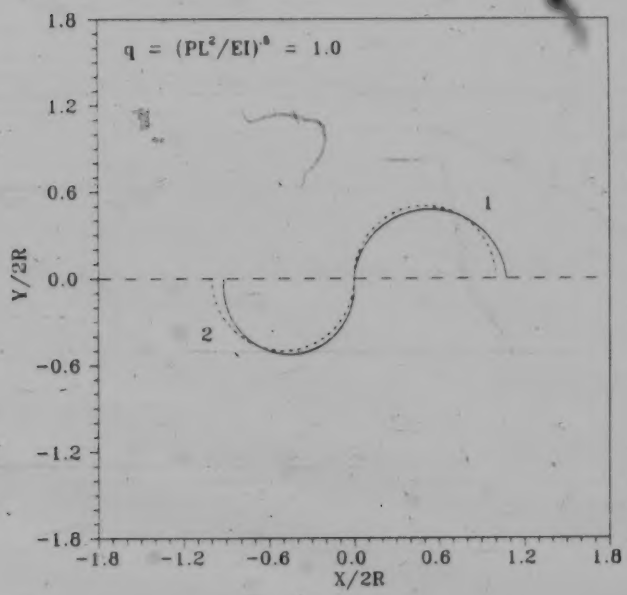


Figure 4.14 Deformed and Undeformed Shape of the Arch for $q = 1.0$

The potential energy is again computed by considering a particular class of kinematically admissible displacements which in this case fulfill the kinematic boundary conditions $X_1 = 0$, $Y_1 = 0$ and $Y_2 = 0$. The first two kinematic constraints are automatically satisfied as initial values. The constraint $Y_2 = 0$ is realized by applying moments at the ends of the rod similar to what was done when evaluating the energy for the no load case. This means that when considering any value of the initial slope γ_1, the shooting technique is used to determine the values of M_1, the initial moment, which results in $Y_2 = 0$. In order to determine a reasonable set of kinematically admissible displacements, care must be exercised when "*artificially*" imposing moments on the arch. Figure 4.15 exhibits the results of what can occur when the moment is incorrectly chosen to begin this search. In this case $\gamma_1 = \pi/4$, however, by assuming an initial guess with different signs leads to an entirely different deformation while still satisfying the kinematic boundary conditions $Y_2 = 0$.

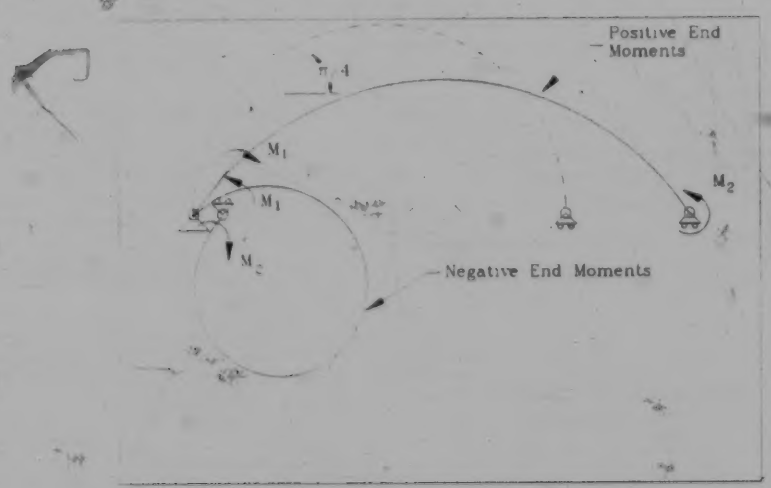


Figure 4.15 Positive and Negative Prescribed Moments at Both Ends Satisfying Boundary Condition $Y_2 = 0$

The strain energy and the corresponding load potential for these two solutions are quite different. Thus if the moments are not applied in a consistent manner, the computed energies may be very much greater than that of the original problem, i.e., the one with $M_1 = M_2 = 0$ and therefore, of not much practical value. If the moments are too excessive, the arch could "*coil*" up forming a spiral which has a very high strain energy due to its high curvature (e.g. see Figure 4.16). Although kinematically admissible displacement fields are still computed, the resulting deformed shapes are very different from the desired shape. As such they are not representative of being "*local*" to the desired solution. Therefore, care must be taken in prescribing the end moments M_1 in order to consider only "*local*" deformed shapes.

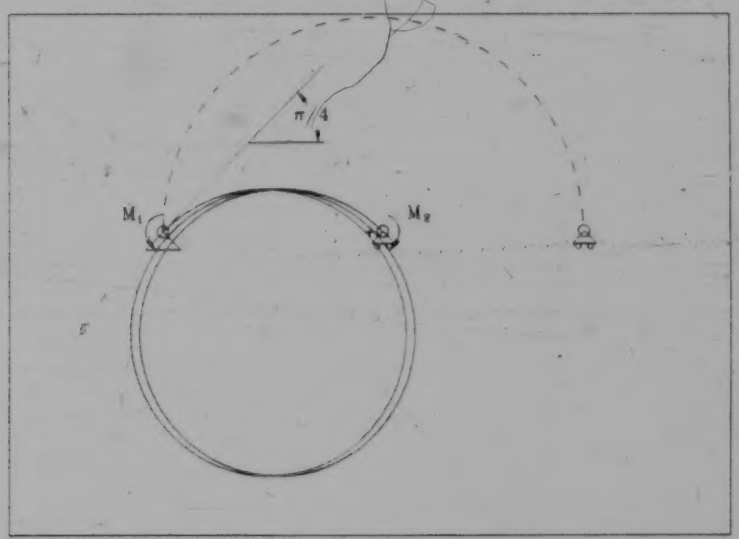


Figure 4.16 Excessive Prescribed Positive Moments Resulting in a "*Spiral*"

By using appropriate moments, the resulting potential energy Π vs γ_1 for $q = 1.0$ is shown in Figure 4.17. Also shown in Figure 4.17 is the potential energy for the

unloaded case (broken line). It is observed that locations of local minimum potential energy exactly correspond to points where $Y_2/2R$ and Λ_2 are zero. Using the minimum potential energy criterion, shape #1 (see Figure 4.14) with an initial slope $\gamma_1 = 1.53$ is globally more stable than shape #2 having an initial slope $\gamma_1 = 4.81$. In order to develop these curves, the shapes and moments used to maintain them were found by starting from the known solutions and working from these to the entire range of initial slopes. This is accomplished by changing the resultant end moment Λ_2 by a small percentage after each iteration using this as initial guess for computing the next small increment of the start slope γ_1 for the next iteration. Maintaining small variations in the estimated moments allows the shooting procedure to generate only "*local*" kinematically admissible displacement fields. The situation where the arch coils up or the arch is "*bent*" the opposite way can then be avoided. As will be indicated later, it is crucial to maintain configurations deformed similarily for identifying multiple solutions which occur for increased applied loads. For this reason, it is difficult to use the potential energy of the rods to find multiple solutions. It can only act as a limited indication of the relative stability of the solutions determined after the solutions are known.

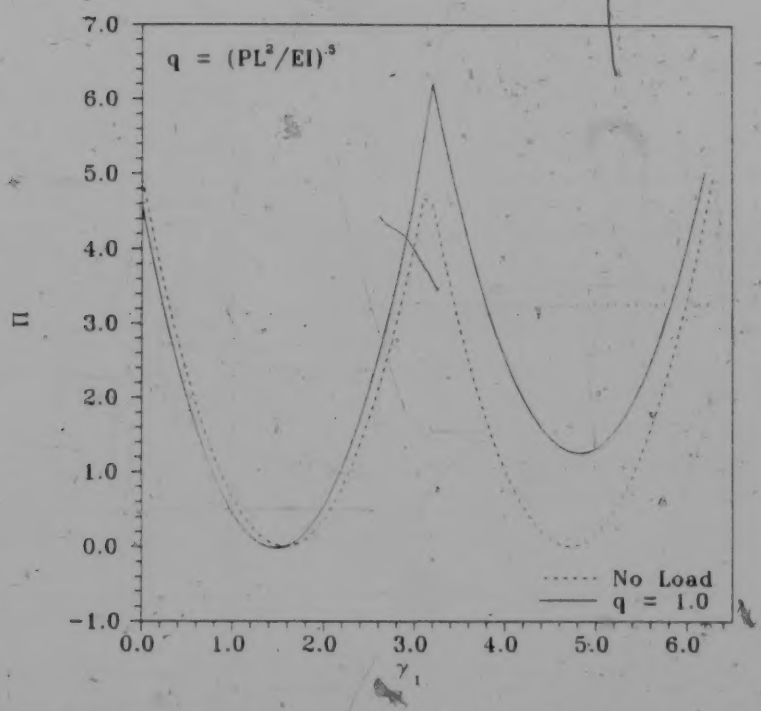


Figure 4.17 Potential Energy Π of Arch for q = 0.0 and 1.0

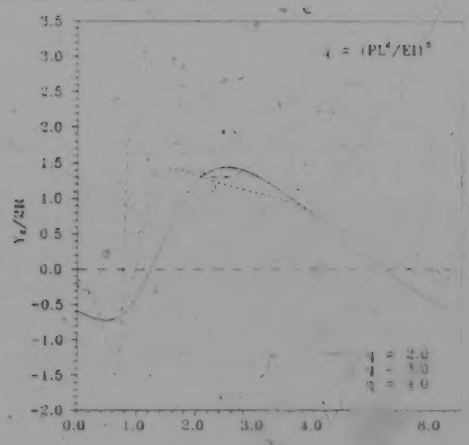


Figure 4.18 $Y_2/2R$ vs Start Slope γ_1 of the Arch for q = 2.0, 3.0 & 4.0.

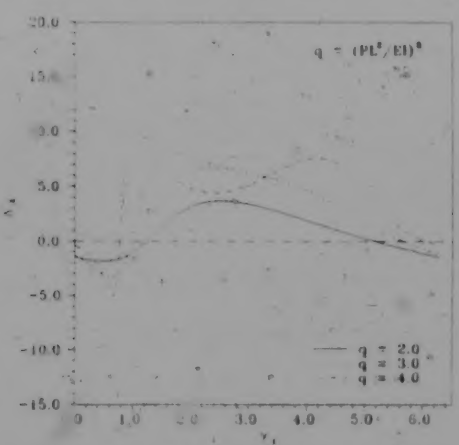


Figure 4.19 Λ_2 vs Start Slope γ_1 of the Arch for q = 2.0, 3.0 & 4.0.

As the load parameter q is increased, more than two solutions are anticipated. To search for these solutions, the shooting technique is again used with either the final y position, Y_2 or the final moment, Λ_2, being the target as the initial slope is varied. Plots of $Y_2/2R$ vs γ_1 and Λ_2 vs γ_1 for q = 2.0, 3.0 & 4.0 are illustrated in Figure 4.18 and Figure 4.19 respectively. Two zero crossings are found for q = 2.0, 3.0 & 4.0 implying only two equilibrium solutions exist. The corresponding deformed geometries are shown in Figure 4.20. Using the same approach, the potential energy of the deformed configurations is computed while satisfying the kinematic boundary conditions. Stability of the arch under various loads q is evaluated by noting that two local minima are found as shown in Figure 4.19. Finally, Table 4.1 summarizes the above findings.

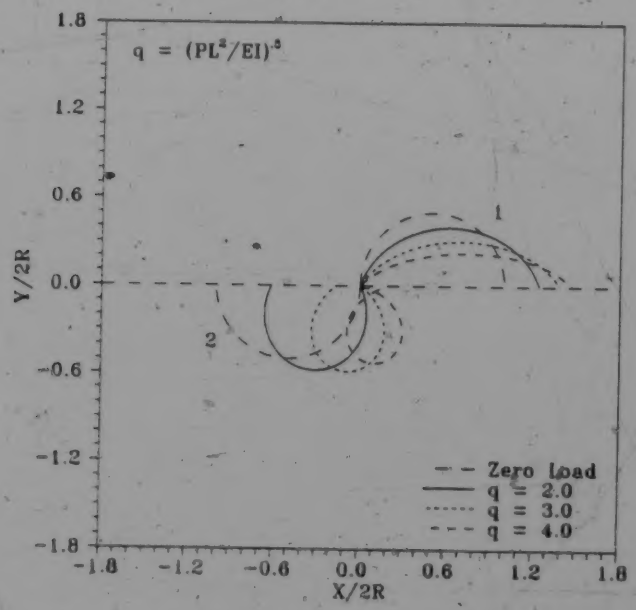


Figure 4.20 Deformed and Undeformed Shapes of the Arch for q = 2.0, 3.0 & 4.0.

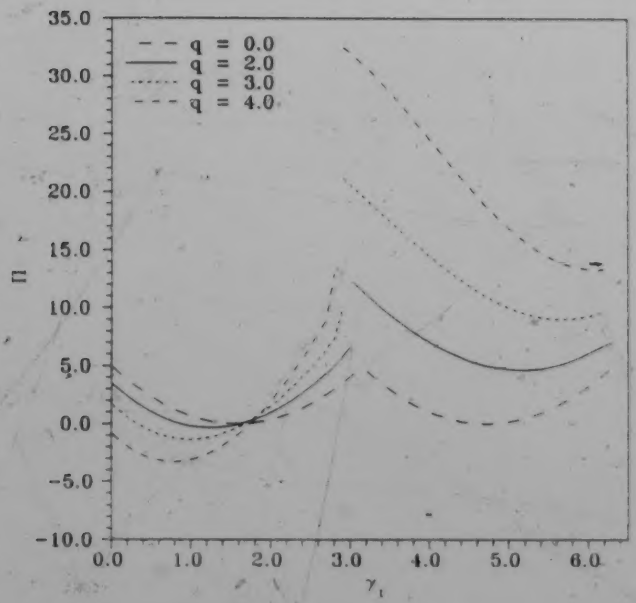


Figure 4.21 Potential Energy Π vs Start Slope γ_1 of the Pin-Roller Support Arch

Table 4.1 Multiple Equilibrium Solutions of the Arch for q = 1.0, 2.0 3.0 and 4.0

q	γ_1 (radians)	Π
1.0	1.5	-0.0229
	4.8	1.248
2.0	1.3	-0.3271
	5.2	4.633
3.0	1.1	-1.315
	5.7	9.031
4.0	0.9	-3.228
	6.1	13.32

For q = 5.0, the plot of $Y_2/2R$ and Λ_2 versus γ_1 facilitates in locating additional multiple solutions as shown in Figure 4.22. Four zero crossings indicating four equilibrium solutions are found. Both kinematic constraint $Y_2/2R = 0$ and force constraint $\Lambda_2 = 0$ are satisfied at these points.

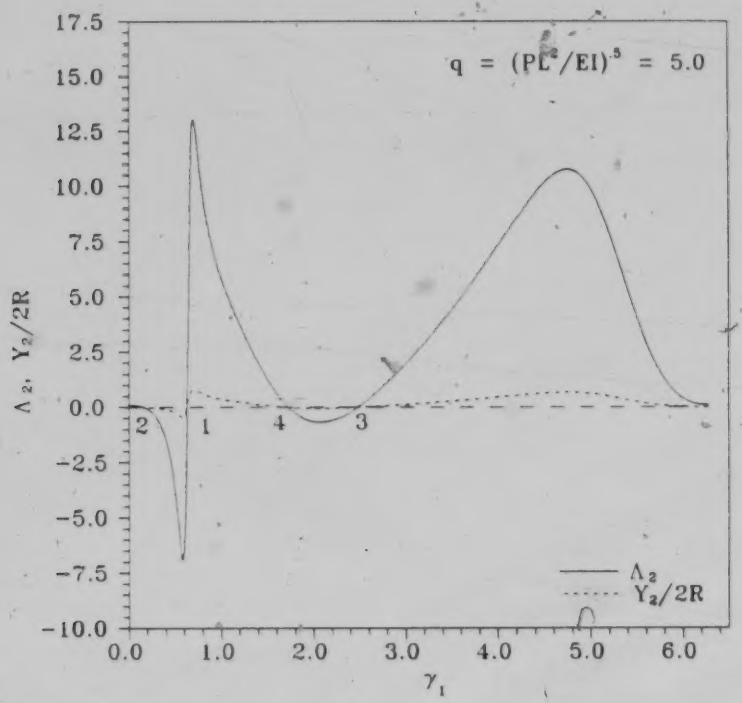


Figure 4.22 Combined End Moment Λ_1, $Y_2/2R$ vs Start Slope γ_1 of the Pin-Roller Support Arch

Table 4.2 Multiple Equilibrium Solutions of the Arch for q = 4.0, 5.0 and 6.0

q	Solution at γ_1						# Sol'n
4.0	0.774	6.039					2
5.0	0.630	0.109	2.472	1.730			4
6.0	0.527	0.103	3.028	1.015	5.973	0.353	6
Shape #	1	2	3	4	5	6	

Table 4.2 summarizes the multiplicity of solutions for q = 4.0, 5.0 and 6.0. Recall that two solutions are present for q ranging from 1.0 to 4.0. As the load is increased to 5.0, first three then at immediately higher loads four equilibrium solutions are found and at q = 6.0, two more solutions have become possible forming a total of six. Each time when bifurcation occurs, first one then two "*new*" shapes are formed.

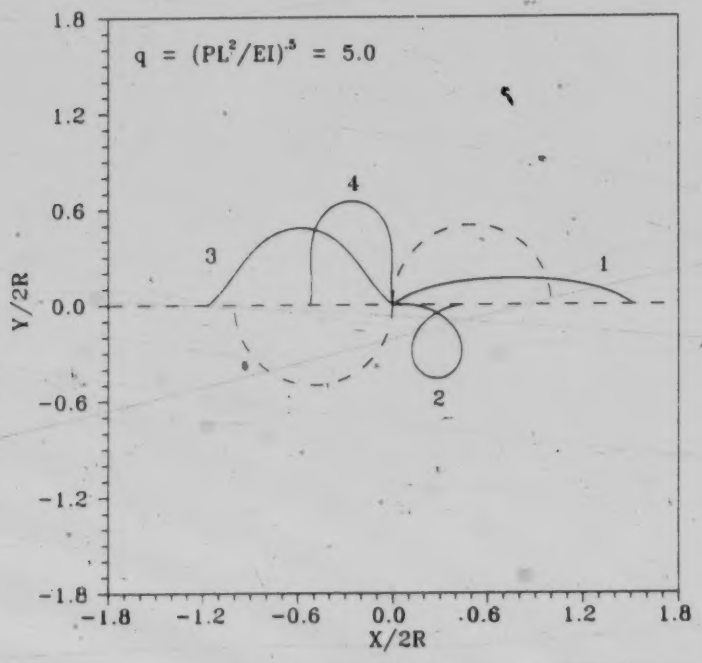


Figure 4.23 Deformed and Undeformed Shapes of Arch for q = 5.0

As shown in Figure 4.23 for q = 5.0, the dashed line indicates the two possible zero solutions whereas the solid lines, labelled #1, #2, #3 and #4, are the deformed configurations and correspond to the shape numbers in Table 4.2. The corresponding potential energy of the deformed configurations are shown in Figure 4.24. Recall that when the potential energy Π of an unloaded arch is investigated, caution must be used

when prescribing end moments to satisfy kinematic boundary conditions. The same approach is taken here to compute the potential energy associated with multiple shapes. Each deformed configuration shown in Figure 4.23 has a different energy level. In order to evaluate the stability of the deformed configurations using the segmental approach, prescribed end moments must be applied in such a way that the deformed configurations retain geometric similarity. The scenario where the arch "*coils*" up due to excessive prescribed moments must be avoided.

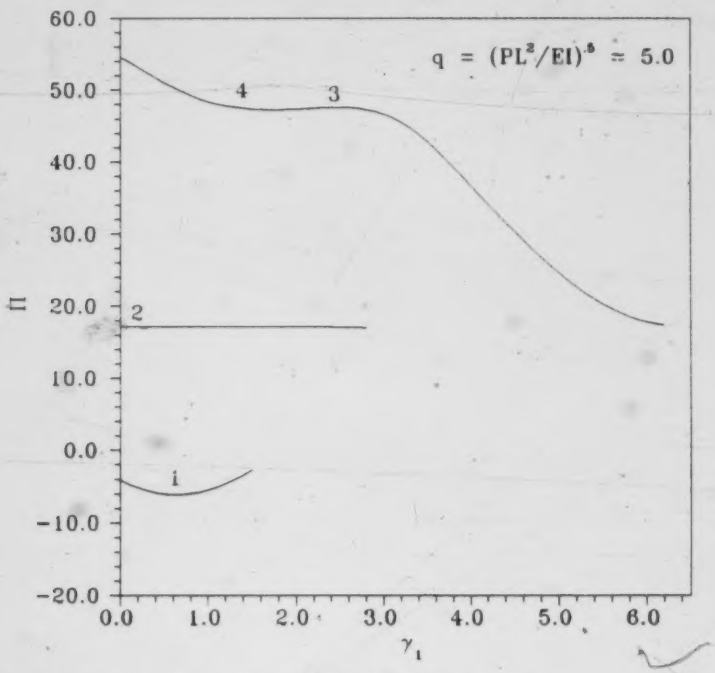


Figure 4.24 Potential Energy Π of Arch for q = 5.0

The local minimum in energy corresponding to the deformed shapes are marked on the plot. It can be seen that shape #1 has the lowest potential energy with shape #2 being the second lowest. This is consistent for the cases of the lower loads discussed before.

That is, shape #1 is at a lower energy and hence more stable configuration than shape #2. Moreover, two new deformed shapes not observed in the lower load case are found with higher energy levels. It must be noted that in the range $4.0 < q < 5.0$, there exists a load q^* such that the arch has only three deformed configurations. At this point shapes #3 and #4 are coalesced. Once the load is increased, the two new shapes (#3 and #4) emerge. These new solutions are at similar energy level as shown in Figure 4.24. The potential energy curve associated with shape #2 is very flat suggesting that shape #2 would require only a small disturbance to perturb it. Therefore, shape #2 exhibits an almost "*neutral stability*" condition.

The fact that the energy curve near shape #2 is exceptionally flat means it is not surprising that further equilibrium shapes of the structure at higher loads occur with shapes and energy levels similar to shape #2. This is shown in Figure 4.25 and 4.26 for $q = 6.0$ where two new shapes, #5 and #6 have developed from the shape at similar energies to #2 (Figure 4.24). It is also noticed that as the load parameter q is increased, the location of local minima in energy for shapes #3 and #4, compared to $q = 5.0$, are at points more widely separated and the difference in energy level between them is increased. It must be emphasized that only a partial determination of the stability of the various equilibrium solutions discussed above has been given. This is due to the fact that all kinematically admissible displacement fields have not been considered. The limited determination of stability for the above equilibrium shapes seems intuitively correct but this may just be fortuitous.

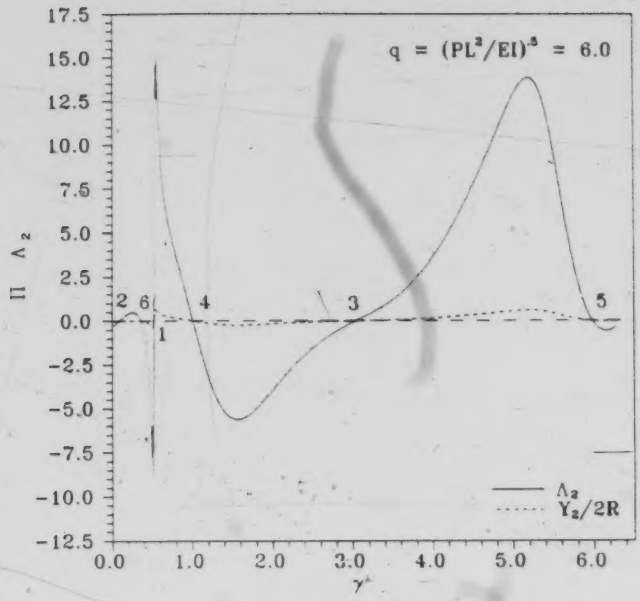


Figure 4.25 Combined End Moment Λ_2, $Y_2/2R$ vs Start Slope γ_1 of the Pin-Roller Support Arch for q = 6.0

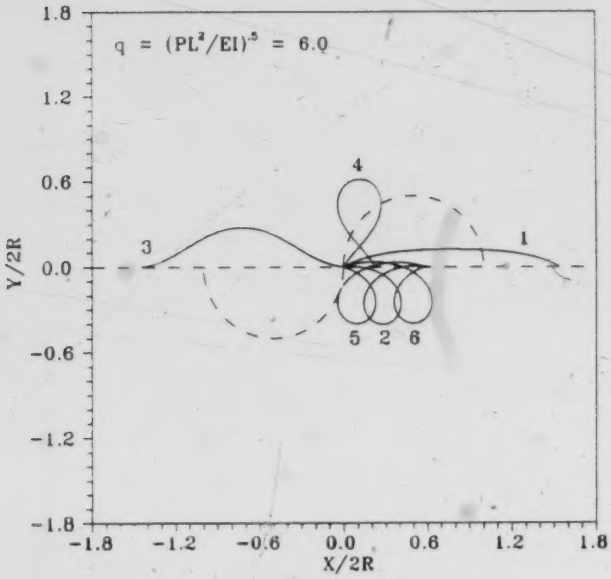


Figure 4.26 Deformed and Undeformed Shapes of Arch for q = 6.0

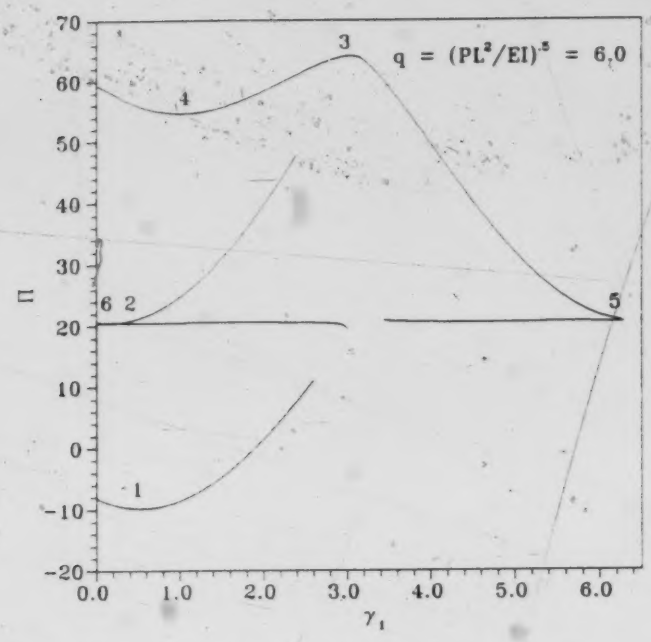


Figure 4.27 Potential Energy Π of Arch for q = 6.0

As mentioned, six deformed configurations are found with #2, #5 and #6 having both similar shapes and energies.

4.3.3 Zero Load Solutions(s) Revisited

Before proceeding to the investigation of the arch loaded by a horizontal concentrated load at the crown, the zero load solutions are reconsidered. Consider again the arch pinned at one end and roller support at the other as shown previously in Figure 4.7. Let the arch assume a deformed configuration and the associated free body diagram is shown in Figure 4.28 with horizontal and vertical reaction forces F_x, F_y applied at both ends regardless of the loading considered.

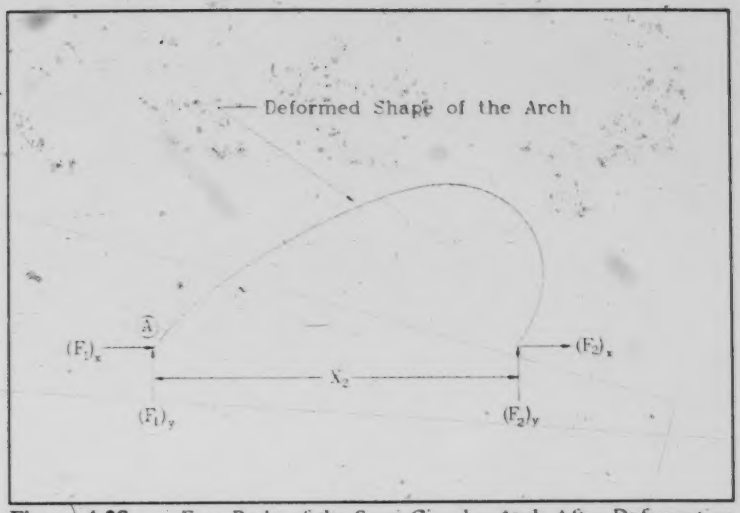


Figure 4.28 Free Body of the Semi-Circular Arch After Deformation

Static

equilibrium on the arch yields,

$$\rightarrow + \sum F_x = 0 = (F_1)_x + (F_2)_x \tag{4.8}$$

$$\uparrow + \sum F_y = 0 = (F_1)_y + (F_2)_y \tag{4.9}$$

$$+ \sum M_A = 0 = -X_2(F_2)_y \tag{4.10}$$

Since a roller support is used at the end of the arch, an additional constraint is found in equation (4.11).

$$(F_2)_x = 0 \tag{4.11}$$

Using equations (4.8) and (4.11) leads to $(F_1)_x = 0$ and considering equations (4.9) and (4.10) results in,

$$0 - -X_2(F_2)_y \rightarrow \begin{cases} (F_2)_y - 0 \\ X_2 - 0 \end{cases} \qquad (4.12)$$

For $(F_1)_y = 0$ and $(F_2)_y = 0$ given by equation (4.9), static equilibrium is satisfied. However, if $X_2 = 0$, then $(F_1)_y = -(F_2)_y$ and they are not necessarily equal to zero. In both cases the equilibrium equations (4.9), (4.10) and (4.11) are satisfied. In other words, if the ends of the arch do not coincide (i.e., $X_2 \neq 0$), $(F_1)_y = (F_2)_y = 0$ whereas $(F_1)_y = -(F_2)_y \neq 0$ for $X_2 = 0$. Since it is assumed that both the forces and deformations are planar, static equilibrium of the arch can be established when the ends coincide after deformation. This means that any arbitrary values of $(F_1)_y$ and $(F_2)_y$ satisfy equilibrium in the y-direction and hence an infinite number of zero load solutions exist. As an illustration, consider an unloaded arch (q = 0.0) evaluated using the segmental approach. However this time, instead of initiating the segmental procedure with initial tension or shear being close to zero (the known solution), a large value is given as the initial assumptions. It is possible that the solution would converge to a deformed configuration in which both ends coincide and the resultant reaction forces cancel each other. Consequently, in addition to the two zero load solutions (#1 and #2) discussed before, any number of other zero load solutions such as #3 and #4 shown in Figure 4.29 can be generated. It is observed that the "*newly found*" zero load solutions are formed with both ends coincident. Contrary to the expected zero horizontal and vertical forces at both ends for an unloaded arch, the vertical forces for zero load at each support solutions #3 and #4 are found to be quite high. Even though the arch retains a deformed

configuration and reaction forces other than the ones expected, static equilibrium conditions are still established. The presence of these zero load solutions poses difficulties when attempting to locate all multiple equilibrium solutions for the arch under a given load.

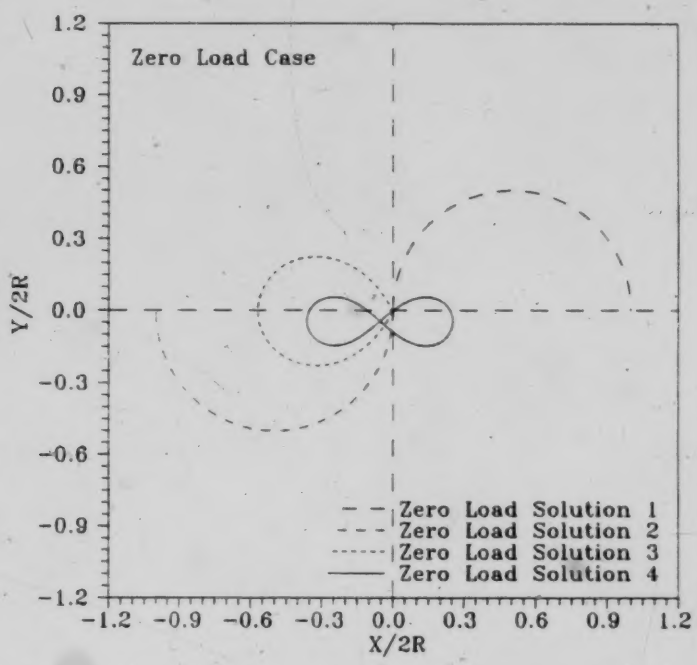


Figure 4.29 Possible Zero Load Solutions for the Pin-Roller Arch

It is noted that when the concentrated load is applied at the roller end, these configurations with both ends coincident do not occur. This is because the tension T_2 and shear V_2 at the roller support are replaced by the applied horizontal load P as shown in Figure 4.30. Performing static equilibrium on the arch leads to

$$V_1 = P\cos\gamma_1 \qquad (4.13)$$

and

$$T_1 = P\sin\gamma_1, \tag{4.14}$$

which are only functions of the start slope γ_1 instead of functions of the forces at the roller end. Therefore, one cannot select arbitrary values of the forces at the roller end such that equilibrium is established. Hence, a finite number of equilibrium solutions are present if the arch is loaded by a horizontal load at the roller support. If the externally applied load is at a position other than the roller support, an infinite number of equilibrium solutions are present because of the possibility of them assuming a deformed equilibrium geometry with both supports coincident.

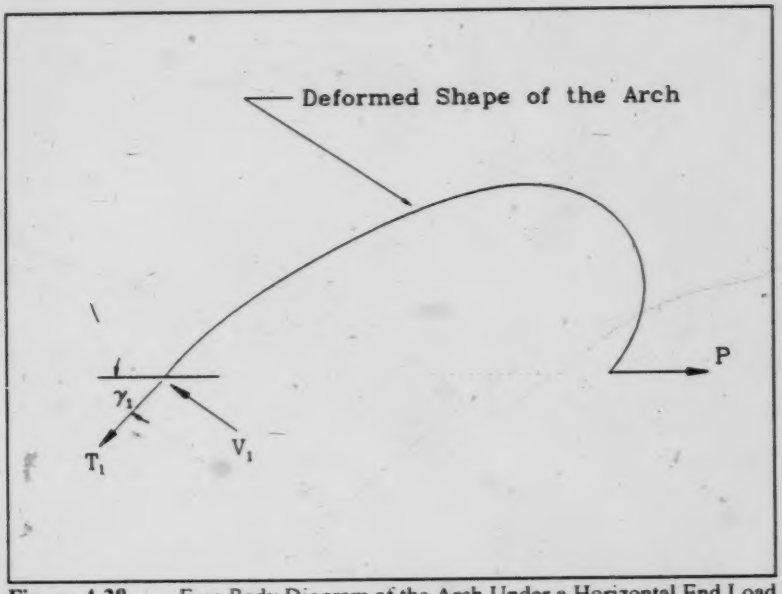


Figure 4.28 Free Body Diagram of the Arch Under a Horizontal End Load

4.3.4 Arch with Concentrated Crown Load

Consider a semi-circular arch subject to a horizontal concentrated load applied at its crown as shown in Figure 4.31.

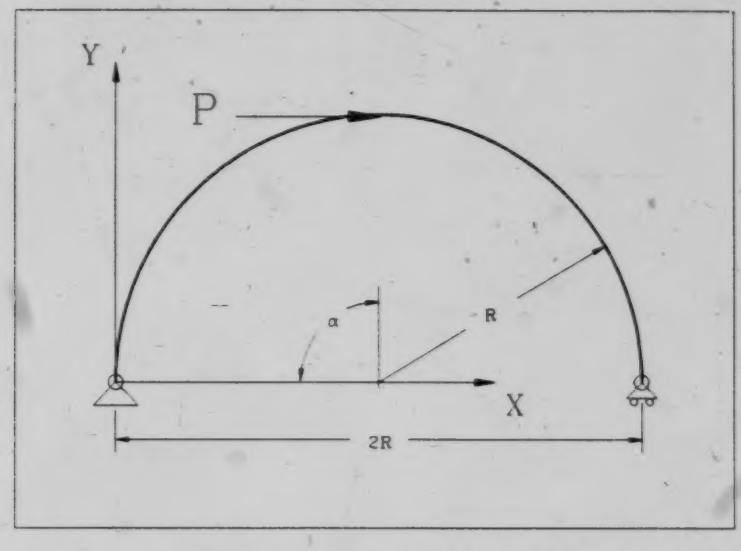


Figure 4.31 Semi-Circular Arch Under a Concentrated Crown Load

Performing static equilibrium along the X-axis and using the fact that the horizontal reaction in the X-direction at the roller support is zero, the tension at the start is,

$$T_1 = \frac{P - V_1 \sin\gamma_1}{\cos\gamma_1} \tag{4.15}$$

As a result, the tension at the start T_1 of the arch is given as a function of the start slope γ_1 and start shear V_1. This enables the problem to be formulated as a two initial unknowns problem where γ_1 and V_1 are the initial unknowns. Because of the discussion in the previous section, it is thus realized that an infinite number of equilibrium solutions

are possible if the ends of the arch are allowed to coincide. When the arch is loaded as shown in Figure 4.31, the known boundary conditions to this problem are;

$$\begin{aligned} X_1 &= 0, & Y_2 &= 0, \\ Y_1 &= 0, & M_2 &= 0, \\ M_1 &= 0, \\ T_1 &= (P - V_1\sin\gamma_1)/\cos\gamma_1. \end{aligned} \tag{4.16}$$

The start shear V_1 and slope γ_1 are the two initial unknowns needed to initiate the segmental procedure. In order to locate all the solutions, an extension of the method for locating multiple solutions for the one variable case is employed. Since there are two initial unknowns and two corresponding known constraints at the end boundary, a data set is generated for initial values γ_1 and T_1 which lead to final values for Y_2 and M_2. A solution to the problem is found when certain values of γ_1 and T_1 lead to the end conditions Y_2 and M_2 equal to zero. The shooting technique is able to search for a solution given initial guesses for γ_1 and V_1, however, the problem is to find **all** the solutions by ensuring that the proper ranges of these input quantities are all evaluated. To this end, contour plots are used to graphically represent the functional dependence. Contours of constant $Y_2 = 0$ and $M_2 = 0$ are superimposed on plots which allow variations of the input variable γ_1 and T_1. A commercial package (see Appendix C) was used to interpolate these data points and form the required contour plots. For an applied load $q = 1.0$, contours of $\Lambda_2 = 0$ and $Y_2/2R = 0$ on γ_1 vs T_1 is shown in Figures 4.32a and 4.32b. Since the denominator in equation (4.15) is zero at $\gamma_1 = 0$ and π, the contours shown in Figure 4.30a and 4.30b are generated with a range of $29\pi/30 < \gamma_1$

$< -\pi/30$ and $-29\pi/30 < \gamma_1 < \pi/30$ to avoid this singularity. Here, the tension is non-dimensionalized as Υ with respect to the length of the arch using

$$\Upsilon = \frac{TL^2}{EI}. \qquad (4.17)$$

Since different contours are superimposed on each plot, a legend is employed with solid line and different length dashed lines denoting the contours of $\Lambda_2 = 0$, $Y_2/2R = 0$ and $X_2/2R = 0$.

Legend:

(Figures 4.32a - 4.36b)

—— $\Lambda_2 = 0$
····· $Y_2/2R = 0$
- - - $X_2/2R = 0$

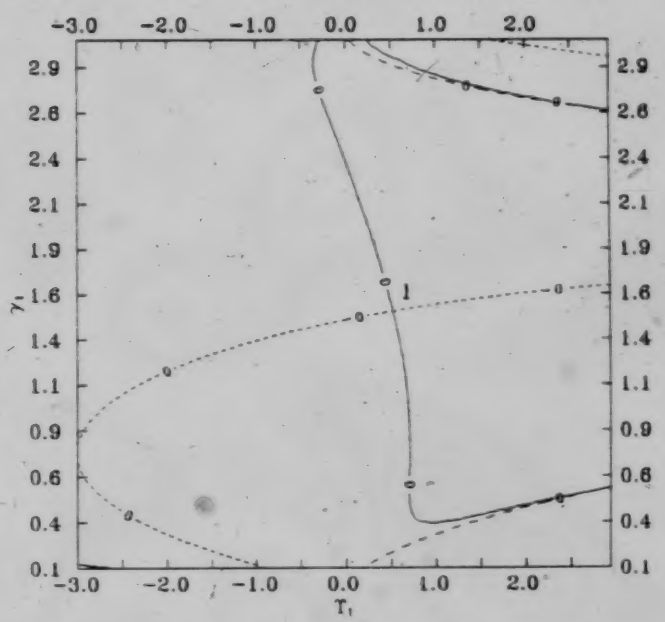


Figure 4.32a Contour Plot for q = 1.0 ($\pi/30 < \gamma_1 < 29\pi/30$)

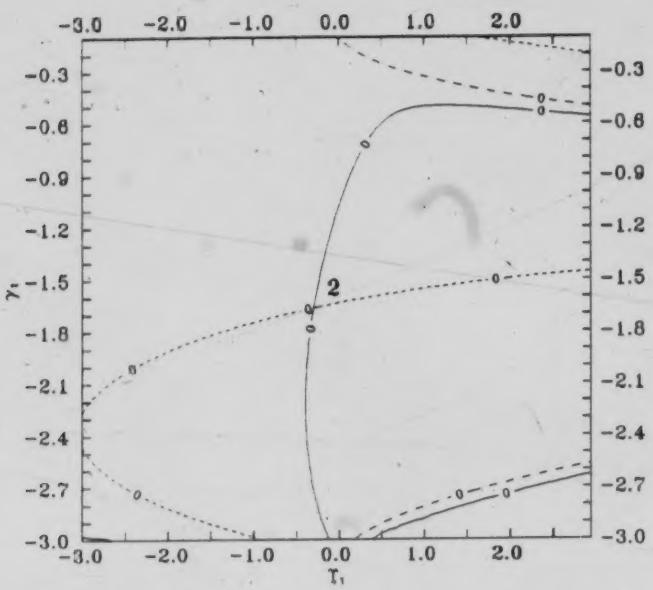


Figure 4.32b Contour Plot for q = 1.0 ($-29\pi/30 < \gamma_1 < -\pi/30$)

A solution to the problem is found when the $\Lambda_2 = 0$ and $Y_2/2R = 0$ contours intersect. The values of γ_1 and T_1 can then be obtained directly from the plot and V_1 is computed using equation (4.15).

In addition to the contours $\Lambda_2 = 0$ and $Y_2/2R = 0$, the contour of $X_2/2R = 0$ is also shown in Figures 4.32a and 4.32b. Since an infinite number of solutions are present when both ends of the arch meet, the additional contour $X_2/2R = 0$ serves as an indication of when the ends of the arch coincide. That is, at the intersections of $\Lambda_2 = 0$, $Y_2/2R = 0$ and $X_2/2R \neq 0$, it is concluded that these two solutions represent desired deformed configurations as shown in Figure 4.33 for q = 1.0.

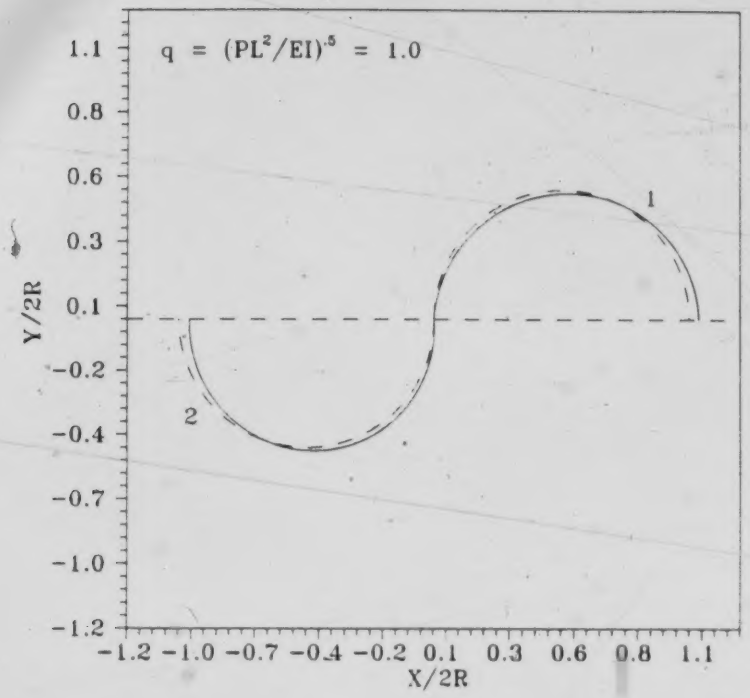


Figure 4.33 Multiple Solutions of the Arch With Crown Load q = 1.0

The range of non-dimensional tension Υ_1 considered is approximately three times the nondimensional applied load q. It is felt that equilibrium of the arch with resultant forces more than plus or minus three times the applied load is possible only when the ends of the arch meet. No definitive argument could be drawn in deciding the range of Υ_1 to be considered. Attempts have been made with a Υ_1 range of more than plus or minus ten times the applied load, however, it was found that the additional solutions beyond the "*three times the applied load*" range are all deformed configurations with both ends coincident. As a result, the range of Υ_1 considered was restricted to a maximum of plus or minus three times the applied load.

The intersection of the different zero contours of constant Λ_2, $Y_2/2R$ and $X_2/2R$

point to the desired solution. The stability of these deformed configurations, however, was not evaluated. Since there are two initial unknowns, the minimum in potential energy associated with a stable configuration would be represented by a minimum point in a three dimensional plot. That is, Π is considered a function of Υ_1 and γ_1. A three dimensional interpretation of the potential energy function is possible but proved to be quite difficult to obtain. Furthermore, when a commercially available surface rendering routine was used, its accuracy was unknown. Efforts have been made to consider cross-sections of the three dimensional surface generated by slicing the three dimensional surfaces and producing plots of Λ_2 vs γ_1 and Λ_2 vs Υ_1 to help locate local minima in the potential energy. If a local minimum was found at the same location for both slices, then a (an actual) minimum point was defined. It should be noted that performing this slicing of a three dimensional surface required a very detailed data set and was quite laborious. Furthermore, visualizing such a minimum is very difficult especially when the potential energy minimum is extremely localized and might be very shallow. This technique is able to quantify the energy of these deformed configurations, however, this approach is of the "*brute force type*" and therefore requires considerable effort.

It should also be noted that representative contours shown in Figures 4.32a and 4.32b require a vast amount of data to generate. A total of 7200 data points per contour were generated requiring two hours of CPU time on an IBM® 80386/387 personal computer. Since a commercial package (see Appendix C) is employed to form the contours, its accuracy and reliability are not known. For numerical efficiency reasons, it was decided to first employed a coarser data set (3600 data points) in such a way that

the contour plots serve only as a good indication of where the solutions occur. From this contour plot, estimations of the initial input to the segmental procedure allowed convergence to more accurate solutions. In all cases, the shooting procedure used in conjunction with the segmental technique converged to the desired solution.

To follow the progression of the solutions, contour plots of Υ_1 vs γ_1 for constant contours of Λ_2, $Y_2/2R$ and $X_2/2R$ for q = 1.2, 1.4 and 1.6 (see Figures 4.34, 4.35 and 4.36). For each of these loads there were only two deformed configurations as shown in Figure 4.37. It should be noticed that as the applied load parameter q is increased, the contours become more difficult to interpret even though there are only two solutions in each case. These more complex patterns are analogous to the more complicated functional forms shown in Figure 4.18 for the arch loaded at its end. In this case the shapes were developing towards ones which led to more solutions of the equilibrium equations.

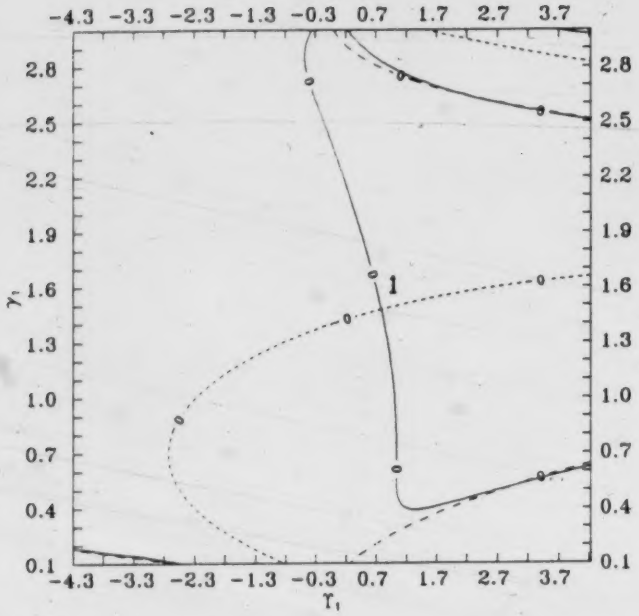


Figure 4.34a Contour Plot for q = 1.2 ($\pi/30 < \gamma_1 < 29\pi/30$)

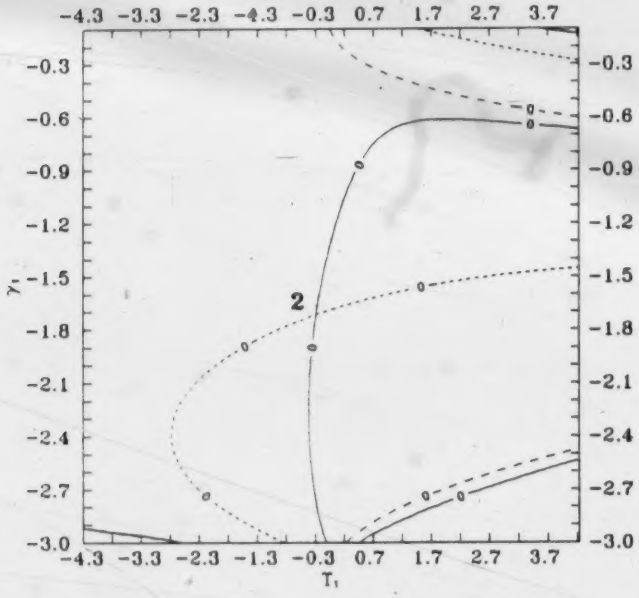


Figure 4.34b Contour Plot for q = 1.2 ($-29\pi/30 < \gamma_1 < -\pi/30$)

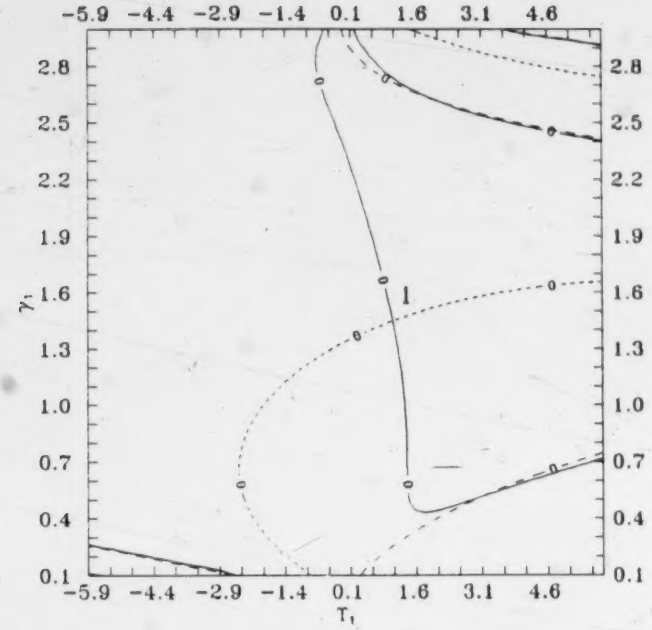


Figure 4.35a Contour Plot for q = 1.4 ($\pi/30 < \gamma_1 < 29\pi/30$)

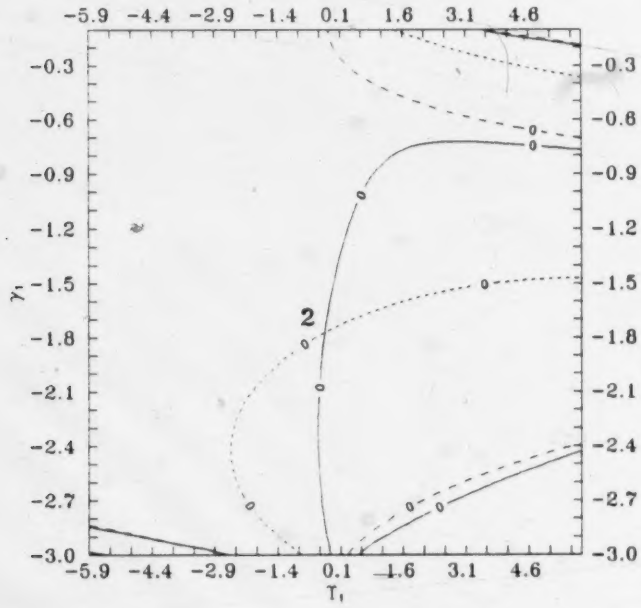


Figure 4.35b Contour Plot for q = 1.4 ($-29\pi/30 < \gamma_1 < -\pi/30$)

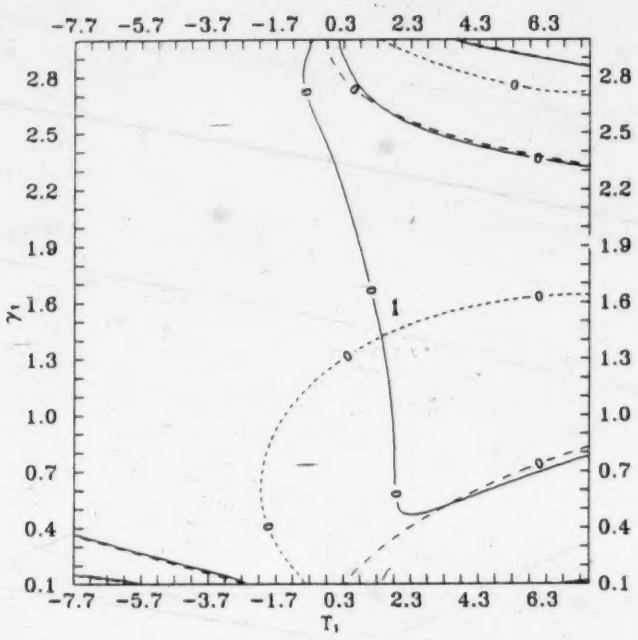


Figure 4.36a Contour Plot for q = 1.6 ($\pi/30 < \gamma_1 < 29\pi/30$)

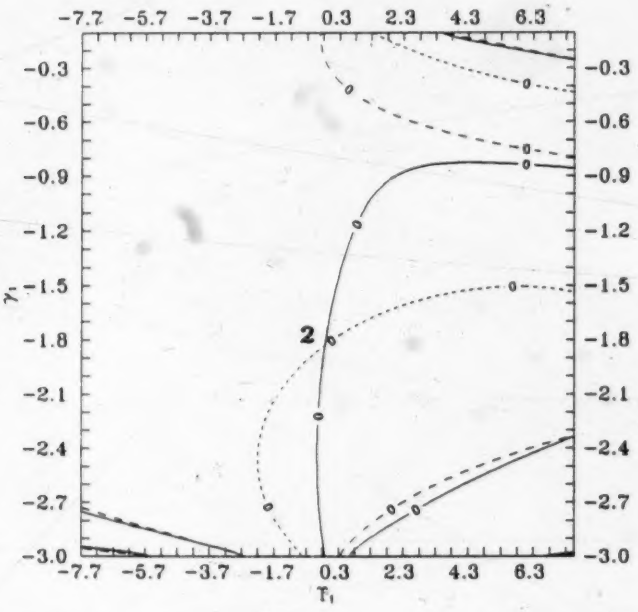


Figure 4.36b Contour Plot for q = 1.6 ($-29\pi/30 < \gamma_1 < -\pi/30$)

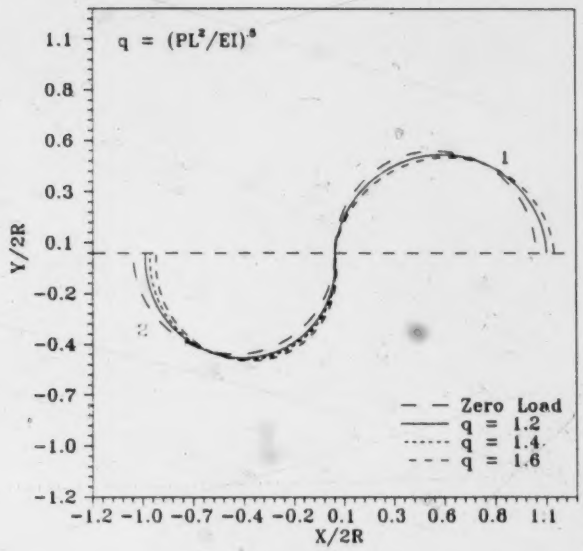


Figure 4.37 Multiple Solutions of the Arch with Crown Load q = 1.2, 1.4 and 1.6

As the applied load is increased, more equilibrium solutions are present as shown in Figure 4.38. At q = 1.8, four intersections of the contours $\Lambda_2 = 0$ and $Y_2/2R = 0$ are found as shown in Figure 4.38a and 4.38b (1, 2, 3, 4). The deformed configurations are shown in Figure 4.39. Note that the ends of the arch are not coincident with each other. Even though #4 appears to have coincident ends, this is not the case. This is further indicated on the contour plots where the contour $X_2/2R = 0$ does not intersect any of the solutions 1, 2, 3 or 4. However, if the applied load parameter q is raised to 2.0, five intersections are found in Figures 4.40a and 4.40b. The intersections of $\Lambda_2 = 0$ and $Y_2/2R = 0$ contour are labelled 1 to 5. It is also seen that for #5, the contour $X_2/2R = 0$ also intersects at that point. As a result, one can conclude that the deformed configuration of the arch corresponding to intersection #5 is one in which both ends of the arch coincide. The resulting deformed configurations are shown in Figure 4.41.

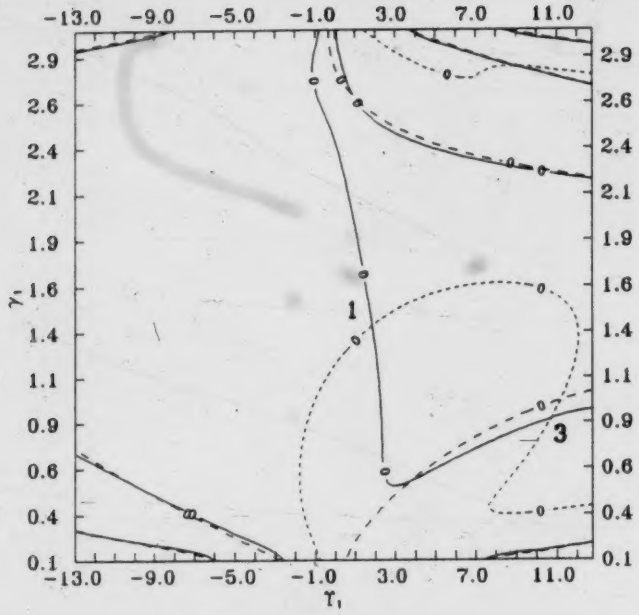


Figure 4.38a Contour Plot for q = 1.8 ($\pi/30 < \gamma_1 < 29\pi/30$)

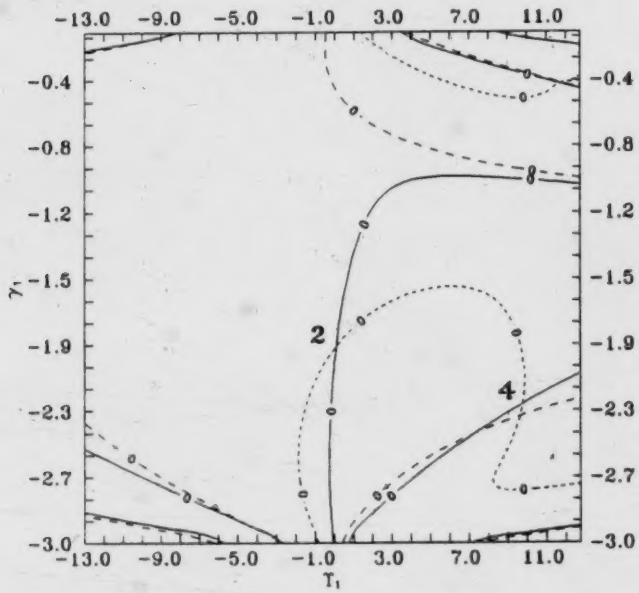


Figure 4.38b Contour Plot for q = 1.8 ($-29\pi/30 < \gamma_1 < -\pi/30$)

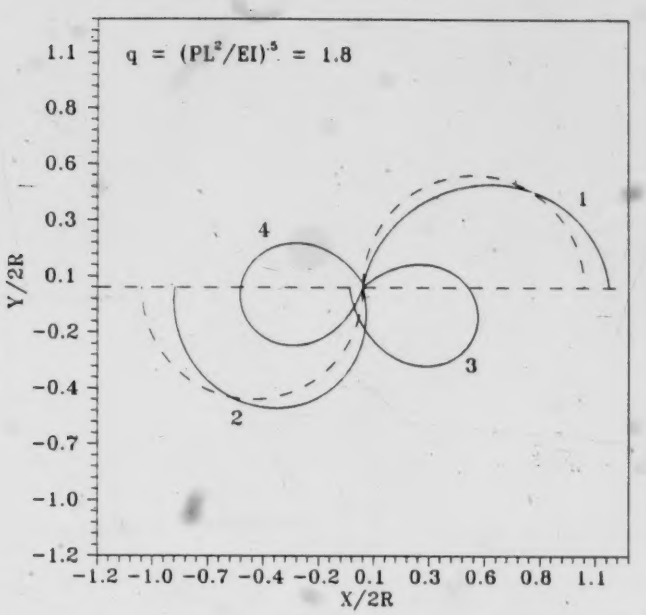


Figure 4.39 Multiple Solutions of the Arch with Crown Load $q = 1.8$

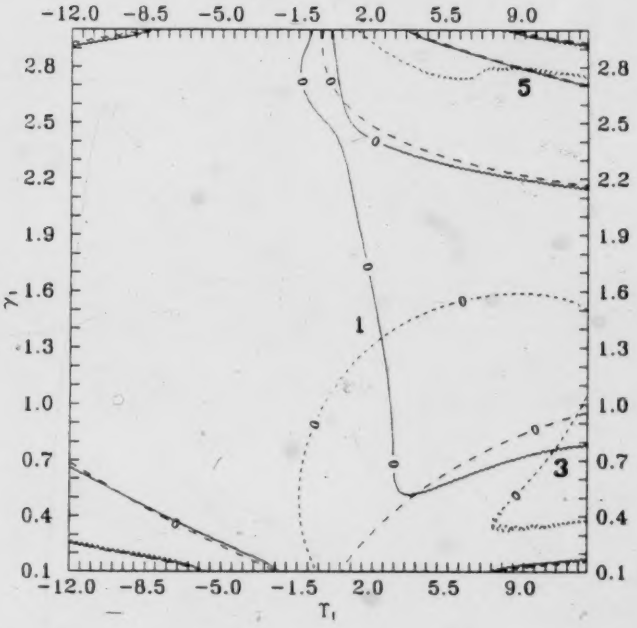


Figure 4.40a Contour Plot for $q = 2.0$ ($\pi/30 < \gamma_1 < 29\pi/30$)

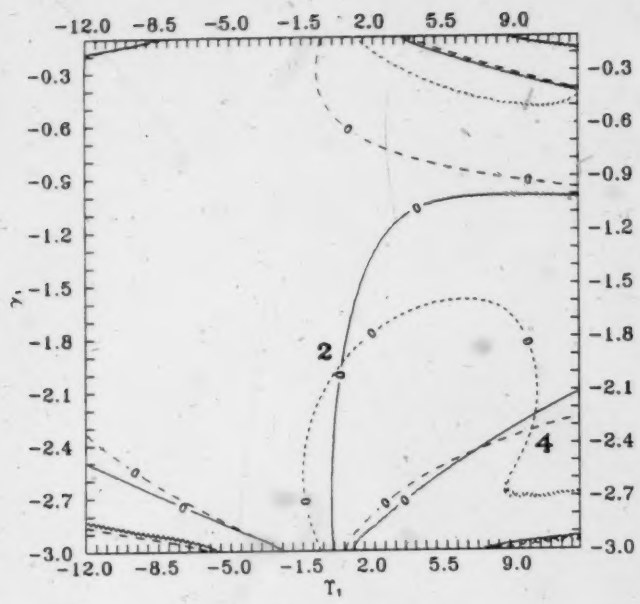


Figure 4.40b Contour plot for q = 2.0 ($-29\pi/30 < \gamma_1 < -\pi/30$)

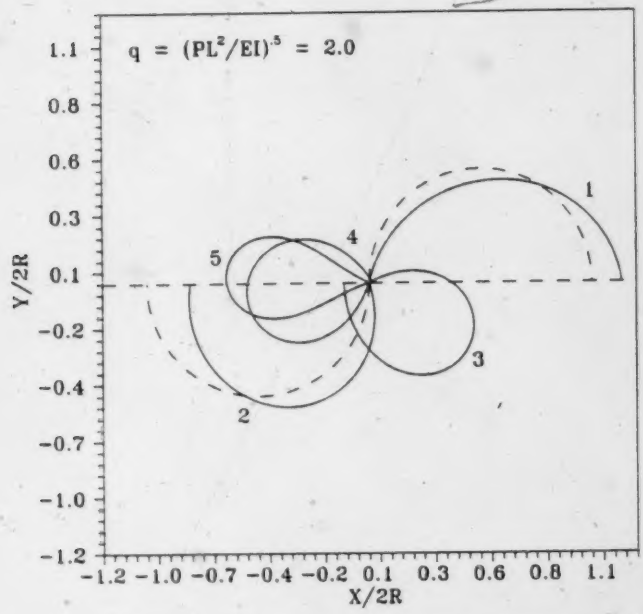
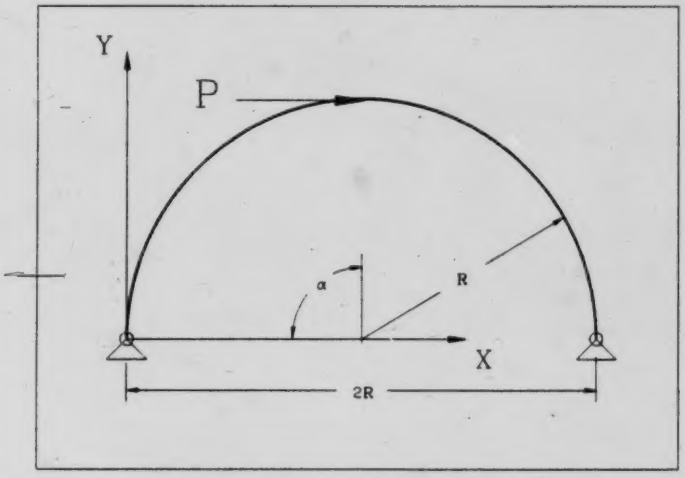


Figure 4.41 Multiple Solutions of the Arch with Crown Load q = 2.0

Deformed shape #4 is also a deformed configuration where both ends of the arch are very close together both do not coincide. This fact is difficult to see in Figure 4.41. It is clearly shown at the intersections of Λ_2 and $Y_2/2R = 0$ in Figure 4.40b and the contour of $X_2/2R = 0$ is extremely close. The actual numerical data shows that the ends are not coincident.

4.4.5 Pin-pin supported Arch with Concentrated Crown Load

Pin-roller supported deep arches under different loadings were considered in previous sections and these generated problems with one and two initial unknown parameters. A three initial unknown problem occurs for a semi-circular arch pinned at both ends and loaded horizontally at its centre (see Figure 4.42). The known boundary conditions of this problem are



$$X_1 = 0,$$
$$X_2 = 2R,$$
$$Y_1 = 0, \qquad (4.18)$$
$$Y_2 = 0,$$
$$M_1 = 0,$$
$$M_2 = 0.$$

Figure 4.42 Semi-Circular Arch Pinned at Both Ends

This leaves the tension T_1, shear V_1 and slope γ_1 as the three initial unknowns. As mentioned previously, with two initial unknowns, visualizing such nonlinear functions in three dimensions posed great difficulties. In this case, in order to find all the solutions using the same brute force method used previously, an extension of three dimension data set would be required. Interpretation of such data is beyond the scope of this study. However, given some reasonable estimates of the initial unknowns, the segmental shooting technique can find some of the multiple solutions. As an example, consider the solutions of the arch as shown in Figure 4.43 for q = 7.69.

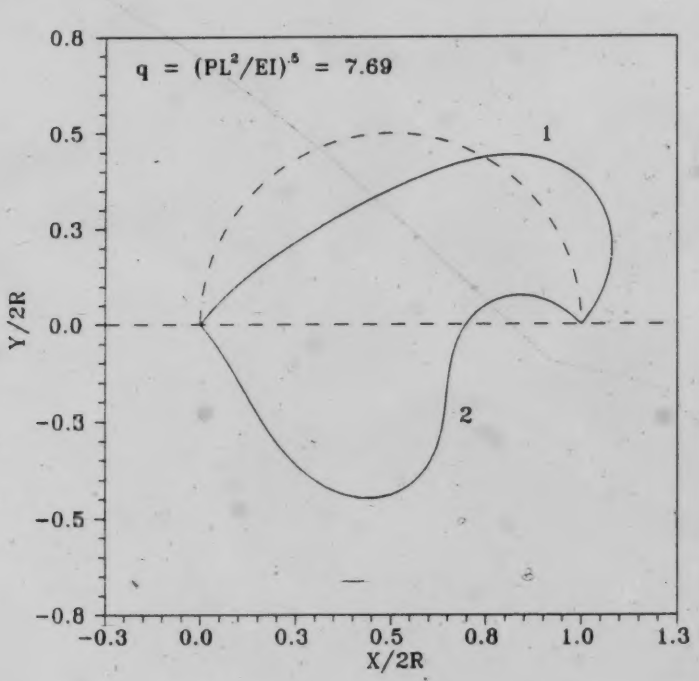


Figure 4.43 Multiple Solutions of the Pin-Pin Arch with Crown Load (α = 90°)

Again, the deformed configurations are labelled #1 and #2 while the undeformed shape is shown in broken lines. Two equilibrium configurations were found. As shown at the

end of Chapter Three, shape #1 was verified with DaDeppo's [19] analytical solution. Furthermore, the segmental approach was able to find one other equilibrium solutions to this problem.

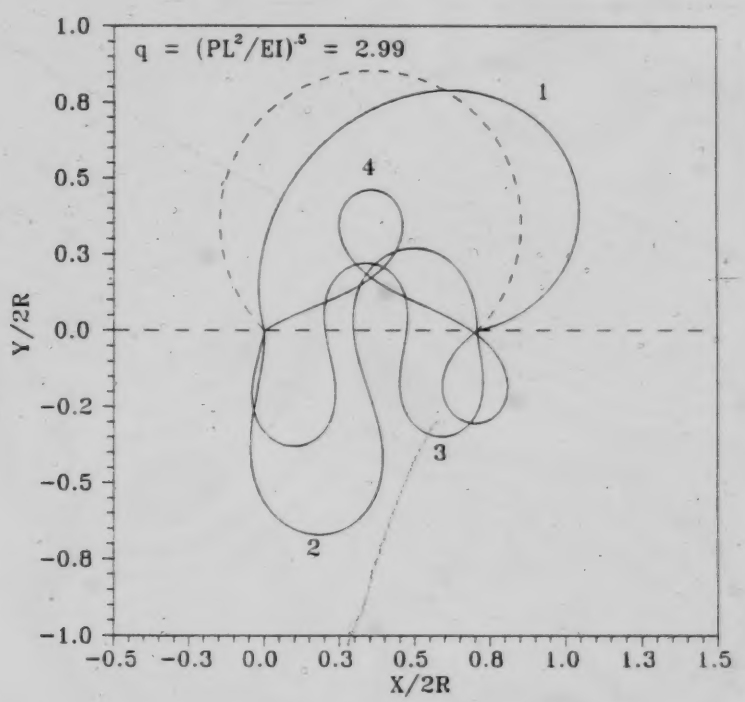


Figure 4.44 Multiple Solutions of the Pin-Pin Arch with Crown Load (α = 135°)

In addition, a pin-pin support arch having a subtending angle α = 135° was also investigated. DaDeppo and Schmidt [19] found a single solution #1 (Figure 4.44) analytically. However, a few other equilibrium solutions were found using the segmental approach in this case (see Figure 4.44). It is not suggested that these are all the solutions, simply, that there are several.

CHAPTER
FIVE

CONCLUDING REMARKS

5.1 Summary

The objective of this research project was to analyse large deflections of various structures including cantilevers and arches. As large deformations are involved, multiple equilibrium configurations are possible for certain loading conditions. The multiplicity of solutions as well as their corresponding stability was considered.

Numerous investigators have examined large deflection rod problems using different analytical and numerical techniques. The segmental shooting technique used in this thesis, however uses a unique technique to analyse nonlinear large deformation rod problems by dividing the rod into numerous small linearized segments. As a result, the nonlinear nature of such problems is directly avoided using a linearized formulation where standard numerical methods are available. Furthermore, this segmental approach converts a boundary value problem into an initial value problem. The problem is then solved by assembling the segments together using force and geometric compatibility conditions. A false position iterative procedure is then employed to find a sequence of solutions which converge to the solution of the boundary value problem.

The segmental shooting technique, when compared to other techniques, has

several features which distinguish it and are worth mentioning. Since the rod of interest is divided into many segments and each segment, being an initial value problem, is treated one at a time, this numerical procedure thus requires minimal computer memory. It can even be executed on a small handheld programmable calculator. Also, the segmental approach is less computationally intensive when compared with, for instance, the finite element formulation. A desktop personal computer with limited memory can optimally execute this procedure. For example, one of the multiple deflected shapes as well as the associated resultant forces and moments of a loaded arch was computed using an IBM® 80386/387 personal computer in less than one minute. Because the problem is posed in terms of initial values it is more adaptable to finding multiple roots of the nonlinear functions. This has been used to advantage in the present study.

As an introduction to the analysis of the multiplicity of solutions to large deformation nonlinear rod problems, the simplest case of a cantilever under various loading was examined. The presence of one initial unknown in the false position procedure characterizes it as a one variable iterative problem. The deformed shapes of the cantilever under free end concentrated load, distributed load and normal load compared favorably with other known solutions. It was found that in problems which have multiple solutions that the initial estimates of the solution needed to be very close to the correct solution in order for the shooting procedure to converge to the desired solution. Interestingly, while multiple solutions are found for the cantilever under concentrated and vertically distributed loads, under a normal load there was only one unique deformed configuration for a given load.

To examine the stability of these solutions, the potential energy was evaluated using the segmental technique and by considering a subset of all kinematically admissible displacement fields. These were all static equilibrium states of the rod. Thus, the potential energy evaluated using this numerical technique provided some insight into the necessary conditions for stability. This allows the shapes to be ranked relatively by examining their corresponding potential energy. It should be noted that this procedure allows large perturbations instead of only infinitesimal variations which have been used in other studies.

After verifying the procedure using cantilever problems, an arch with pin support at one end and roller support at the other under horizontal concentrated load at the crown or at the roller support is considered. It was noticed that if the arch is pin-roller supported, there exist an infinite number of solutions when the two ends of the arch coincided. When the concentrated load is applied at the roller support, the problem is essentially a one initial unknown problem where a solution technique similar to that used for the cantilevers is applicable. When the applied load is at the crown of the arch, the problem becomes one with two initial unknowns and contour plots instead of a graph are necessary to locate all the solutions. Due to the vast amounts of data necessary to produce a representative contour, a coarse data set was first made to provide an initial approximation of the solutions. This initial approximation is then used as input into the segmental shooting technique to obtain a precise final solution. In all cases, the estimates obtained from the contour plots were able to allow the segmental shooting technique to converge to the solution. The stability of the deformed configurations was not evaluated

for the two initial unknown problem. This is because such evaluation requires a three dimensional visualization of a minimum point.

The contour method is not feasible with more than two initial unknowns. As a result, deformed configurations of a pin-pin arch under various loads were only presented to illustrate the capability of this numerical technique. No immediate conclusion can be drawn in terms of finding all the multiple solutions.

In summary, the segmental shooting technique was used not only to analyse large deflection elastica problems, but also to evaluate the multiplicity and stability of these solutions. Even though it is somewhat limited when interpreting the stability of these solutions, it appears superior to other methods used to solve problems of this class. This thesis has illustrated that the segmental shooting technique is able to solve large deflections in much generality. These include arches with different support conditions and various loading conditions. Despite the fact that only horizontal concentrated load arch problems are examined, the segmental shooting technique is capable of solving problems under distributed and normal loads as shown in the former chapter. Moreover, boundary value problems with one, two and three initial unknowns have been examined with relative ease.

5.2 Recommendations

Although it is demonstrated that large deflections of inextensible planar rods can be solved efficiently using the segmental shooting technique, continual enhancement on the interpretation of these nonlinear functions involving more than one initial unknown

needs to be done. This includes an improved iterative shooting method for finding roots and a more flexible and efficient visualization of the function in multi-dimensions.

It is felt that an extension of this technique for solving large deflection rod problems in three dimensions would be useful for studying out of plane deformations. However, achieving such an extension will require development of a more efficient numerical iterative shooting procedure when more than three variables are involved.

- REFERENCES -

1. Schmidt, R. and DaDeppo, D.A. "A Survey of Literature on Large Deflections of Nonshallow Arches. Bibliography of Finite Deflections of Straight and Curved Beams, Rings, and Shallow Arches," *The Journal of Industrial Mathematics Society*, Vol. 21, Part 2, 1971.

2. Gorski, W. and Aust, M.I.E. " A Review of Literature and a Bibliography on Finite Elastic Deflections of Bars," *Civil Engineering Transactions*, 1976, pp. 74-85.

3. Wang, C.Y. "A Critical Review of the Heavy Elastica," *International Journal of Mechanical Sciences*, Vol 28, No.8, 1986, pp. 549-559.

4. Frisch-Fay. *Flexible Bars*, Butterworths, London, 1962.

5. Wang, T.M. "Nonlinear Bending of Beams with Uniform Distributed Loads," *International Journal of Nonlinear Mechanics*, Vol. 4, 1969, pp. 389-395.

6. Obata, M., Yoshimitsu, T. and Goto, Y. "Elliptic Integral Solutions of Plane Elastica with Axial and Shear Deformations," *International Journal of Solids and Structures*, Vol. 26, No. 4, 1990, pp. 375-390.

7. Mattiasson, K. "Numerical Results from Large Deflection Beam and Frame Problems Analysed by Means of Elliptic Integrals," *Short Communications*, John Wiley and Sons, 1980, pp. 145-153.

8. Mitchell, T.P. "The Nonlinear Bending of Thin Rods," *Journal of Applied Mechanics*, March 1959, pp. 40-43.

9. Conway, H.D. and Seames, A.E. "A Numerical Procedure for Calculating the Large Deflections of Straight and Curved Beams," *Journal of Applied Mechanics*, June 1957, pp. 289-294.

10. Faulkner, M.G. and Stredulinsky, D.C. "Nonlinear Bending of Inextensible Thin Rods Under Distributed and Concentrated Loads," *Transactions of the CSME*, Vol. 4, No. 2, 1976-77, pp. 77-81.

11. Surana, K.S. "Geometrically Nonlinear Formulation for Two Dimensional Curved Beam Elements," *Computers and Structures*, Vol. 17, No. 1, 1981, pp. 105-114.

12. Surana, K.S. "Geometrically Nonlinear Formulation for the Curved Shell Elements," *International Journal for Numerical Methods in Engineering*, Vol. 19,

1983, pp. 581-615.

13. Surana, K.S. and Sorem, R.M. "Geometrically Nonlinear Formulation for Three Dimensional Curved Beam Elements with Large Rotations," *International Journal for Numerical Methods in Engineering*, Vol. 28, 1989, pp. 43-73.

14. Somervaille, I. and Ngiam, P.C.A. "Intrinsic Coordinate Elements for Elastica Problems," *10th ACMSM University of Auckland* 1987.

15. Somervaille, I. "High Order Transfer Matrices and Applications," *11th ACMSM University of Auckland* 1988, pp. 101-104.

16. Conway, H.D. and Lo, C.F. "The Elastic Stability of Curved Beams," *International Journal of Mechanical Sciences*, Vol. 9, 1967, pp. 527-538

17. Conway, H.D. and Lo, C.F. "Further Studies on the Elastic Stability of Curved Beams," *International Journal of Mechanical Sciences*, Vol. 9, 1967, pp. 707-718.

18. DaDeppo, D.A. and Schmidt, R. "Sidesway Buckling of Deep Circular Arches Under a Concentrated Load," *Journal of Applied Mechanics*, June 1969, pp. 325-327.

19. DaDeppo, D.A. and Schmidt, R. "Large Sideward Deflections of Two Hinged Circular Arches," *Journal of Engineering for Industry*, November 1971, pp. 1268-1274.

20. Huddleston, J.V. "Finite Deflections and Snap-Through of High Circular Arches," *Journal of Applied Mechanics*, December 1968, pp. 763-769.

21. Fried, I. "Stability and Equilibrium of the Straight and Curved Elastica - Finite Element Computation," *Computer Methods in Applied Mechanics and Engineering 28*, 1981, pp. 49-61.

22. Navaee, S. *Alternate Equilibrium Configurations of Flexible Cantilever Beams*, Ph.D. Thesis, Clemson University December 1989.

23. Nordgren, R.P. "On Finite Deflection of An Extensible Circular Ring Segment," *International Journal of Solids and Structures*, Vol. 2, 1966, pp. 223-233.

24. Trahair, N.S. and Papangelis, J.P. "Buckling of Monosymmetric Arches Under Point Loads," *Engineering Structures*, Vol. 10, October 1988, pp. 257-264.

25. Banan, M.R., Farshad, M. and Karmi, G. "Spatial Buckling of Arches - A Finite

Element Analysis," *Computers and Structures*, Vol. 34, No. 4, 1990, pp. 565-576.

26. Antman, S. "General Solutions for Plane Extensible Elasticae Having Nonlinear Stress-Strain Laws," *Quarterly Journal of Applied Mathematics*, Vol. 26, 1968-69, pp. 35-47.

27. Miranda, I., Hughes, T.J.R., Franca, L.P. and Loula, A.F.D. "Stability, Convergence and Accuracy of a New Finite Element Method for the Circular Arch Problem," *Computer Methods in Applied Mechanics adn Engineering 63*, 1987, pp. 281-303.

28. Wang, C.M. and Goh, C.J. "Generalized Shooting Method for Elastic Stability Analysis and Optimization of Structural Members," *Computers and Structures*, Vol. 38, No.1, 1991, pp. 73-81.

29. Thurston, G.A. "Continuation of Newton's Method Through Bifurcation Points," *Journal of Applied Mechanics*, September 1969, pp. 425-430.

30. Reaser, M.H., Kamat, M.P. and Watson, L.T. "A Robust Hybird Algorithm for Computing Multiple Equilibrium Solutions," *Engineering Computers*, Vol. 2, March 1985, pp. 30-34.

31. Tabarrok, B. and Farshad, M. "On Stability Equations of Spatial Rods - A Technical Theory," *Transactions of CSME*, Vol. 11, No. 4, 1987, pp. 229-235.

32. Dahlquist, G. and Björck, A., *Numerical Methods*, Prentice-Hall, 1974.

33. El-Rayes, K. *Numerical Analysis of Orthodontic Appliances*, M.Sc. Thesis, University of Alberta, 1989.

34. Knops, R.J. and Wilkes, E.W. "Theory of Elastic Stability", *Handbuch Der Physik, Mechanics of Solids III*, Vol. VIa/3, 1973, pp. 125 - 302.

APPENDIX
A

ANALYTICAL SOLUTIONS OF THE CANTILEVER UNDER FREE END CONCENTRATED LOAD

In order to verify the segmental shooting technique, knowns solutions were used. For a horizontal cantilever under free end vertical concentrated load, Frisch-Fay's [4] analytical solutions in terms of elliptic integrals of the first and second kind were employed. A bisection root finding procedure was implemented in Microsoft QuickBASIC® to numerically computed this analytical solutions. The following details the source listing.

```
DECLARE SUB ParameterP (p#, q#, TOL#)
DECLARE FUNCTION Fct# (q#, p#, CA#)
DECLARE SUB ModulusP (p#, q#, TOL#)
DECLARE SUB SlopePHI (s#, p#, PhiBDeg#, PhiDeg#, k#, TOL#)
DECLARE FUNCTION ELLIP1# (p#, PHI1#, CA#)
DECLARE FUNCTION CELLIP1# (p#, CA#)
DECLARE FUNCTION ASin# (A#)
DECLARE FUNCTION CELLIP2# (p#, CA#)
DECLARE FUNCTION ELLIP2# (p#, PHI1#, CA#)

'*****************************************************************************************************************

        This program is designed to evaluate the final geometry of a horizontal cantilever beam subjected to a vertical point load
        at the end using the analytical solution detailed in "Flexible Bars" by Frisch Fay.

        The program would be required to evaluate Elliptic Intergrals of the First and Second Kind. The Elliptic Integrals were
        computed numerically using an infinite series expansion detailed in "Numerical Recipies".

                                        Version 1.0  Victor Tam  July 6, 1990

'*****************************************************************************************************************

DEFDBL A-Z

CLS : COLOR 15, 1

'*** Define Constants

PI = 3.1415926535897793#
NumSeg% = 750              'No. of STEPS used
L1 = 150                   'Length of Beam
E = 3E+07                  'Young's Modulus
I = 1 / 12 * 2.5 * .25 ^ 3  'Cross Section Moment of Inertia
TOL = 1E-12                'Numerical Tolerence
CA = .0001                 'Sqr(Accuracy) for Elliptic Integrals
Phi = 0

'***  Compute Parameters

EI = E * I                          'Flexural Rigidity

FOR q = 1 TO 8 STEP .5

   OutFile$ = "d:\qb\qbsegtec\data\Frisch" + RIGHT$(STR$(q * 10), LEN(STR$(q * 10)) - 1) + ".Out"
   OPEN OutFile$ FOR OUTPUT AS #1
   F = q ^ 2 * EI / (L1 ^ 2)
   k = SQR(F / EI)
   ParameterP p, q, TOL
   h = 2 * p / k
   SI0 = ASin((p ^ 2) * 2 - 1)              'in Radians
   PhiB = ASin(1! / (p * SQR(2!)))          'in Radians
   Delta = L1 - h * COS(PhiB)
   PRINT #1, OutFile$
   PRINT #1, "Time Started: ", TIME$, DATE$
   PRINT #1,
   PRINT #1, "q = ", q
   PRINT #1, "p = ", p
   PRINT #1, "F = ", F; "lbs"
   PRINT #1, "E = ", E; "lb/in^2"
   PRINT #1, "I = ", I; "in^4"
   PRINT #1, "EI = ", EI; "lb.in^2"
```

```
    PRINT #1, "Length = ", L1; "in"
    PRINT #1, "k = ", k
    PRINT #1, "h = ", h
    PRINT #1, "PhiB = ", PhiB
    PRINT #1, "Delta = ", Delta
    PRINT #1, "Tolerence = ", TOL

    OPEN "d:\qb\qbsegtec\data\Frisch" + RIGHT$(STR$(q * 10), LEN(STR$(q * 10)) - 1) + ".Gem" FOR OUTPUT AS #2

'*  Vary the position of the arc s along the beam and evaluate the corresponding Phi at s.

    FOR s = 0 TO 149.8 STEP L1 / NumSeg%

      SlopePHI s, p, PhiB, Phi, k, TOL
      X! = h * (COS(PhiB) - COS(Phi))
      Y! = (2 * ELLIP2(p, PhiB, CA) - ELLIP1(p, PhiB, CA) + ELLIP1(p, Phi, CA) - 2 * ELLIP2(p, Phi, CA)) / k
      PRINT X!, Y!, s
      ss! = s
      PRINT #2, X!, -Y!, ss!

    NEXT s

    PRINT #1, "X = ", X!; " in"
    PRINT #1, "Y = ", -Y!; " in"
    PRINT #1, "Phi = ", Phi
    PRINT #1, "Time Completed: ", TIME$, DATE$
    CLOSE
    SOUND 8000, 1
NEXT q

SOUND 4000, 1

END                                    '*** End MAIN ***

FUNCTION ASin (A)

'*****************************************************************************************
'          A function to compute the ArcSine of a Number in terms of ArcTangent
'*****************************************************************************************

    ASin = ATN(SQR(A ^ 2 / (1 - A ^ 2)))

END FUNCTION

FUNCTION CELLIP1 (p, CA)

'*****************************************************************************************
'          A function to compute the Complete Elliptic Integral of the First Kind
'*****************************************************************************************

  QQC = SQR(1 - p ^ 2)
  QC = ABS(QQC)
  PI = 3.1415926535897#
  A = 1
  B = 1
  pp = 1
  E = QC
  EM = 1
1 F = A
  A = A + B / pp
  G = E / pp
```

```
  B = B + F * G
  B = B + B
  pp = G + pp
  G = EM
  EM = QC + EM
  IF ABS(G - QC) > G * CA THEN
     QC = SQR(E)
     QC = QC + QC
     E = QC * EM
     GOTO 1
   END IF
   CELLIP1 = (PI / 2) * (B + A * EM) / (EM * (EM + pp))
END FUNCTION

FUNCTION CELLIP2 (p, CA)

'*****************************************************************************
          A function to compute the Complete Elliptic Integral of the Second Kind
'*****************************************************************************

  QQC = SQR(1 - p ^ 2)
  QC = ABS(QQC)
  PI = 3.1415926535897793#
  A = 1
  B = QQC'B = 1
  pp = 1
  E = QC
  EM = 1
3 F = A
  A = A + B / pp
  G = E / pp
  B = B + F * G
  B = B + B
  p = G + pp
  G = EM
  EM = QC + EM
  IF ABS(G - QC) > G * CA THEN
     QC = SQR(E)
     QC = QC + QC
     E = QC * EM
     GOTO 3
   END IF
   CELLIP2 = (PI / 2) * (B + A * EM) / (EM * (EM + pp))


END FUNCTION

FUNCTION ELLIP1 (p, PHI1, CA)

'*****************************************************************************
          A function to compute the Elliptic Integral of the First Kind
'*****************************************************************************
'
' The desired accuracy is the square of CA.
'
  PI = 3.1415926535897793#
  CB = .01 * CA ^ 2
  QQC = SQR(1 - p ^ 2)
  X = TAN(PHI1)
  IF X = 0 THEN
     ELLIP1 = 0
```

```
ELSEIF QQC < > 0 THEN
   QC = QQC
   A = 1
   B = 1
   C = X ^ 2
   D = 1 + C
   pp = SQR((1 + QC ^ 2 * C) / D)
   D = X / D
   C = D / (2 * pp)
   Z = A - B
   EYE = A
   A = .5 * (B + A)
   Y = ABS(1 / X)
   F = 0
   L = 0
   EM = 1
   QC = ABS(QC)
2  B = B + EYE * QC
   E = EM * QC
   G = E / pp
   D = D + F * G
   F = C
   EYE = A
   pp = pp + G
   C = .5 * (D / pp + C)
   G = EM
   EM = EM + QC
   A = .5 * (B / EM + A)
   Y = Y - E / Y
   IF Y = 0 THEN Y = SQR(E) * CB
   IF ABS(G - QC) > CA * G THEN
      QC = SQR(E) * 2
      L = L + L
      IF Y < 0 THEN L = L + 1
      GOTO 2
   END IF
   IF Y < 0 THEN L = L + 1
   E = (ATN(EM / Y) + PI * L) * A / EM
   IF X < 0 THEN E = -E
   ELLIP1 = E + C * Z
ELSE
   PRINT "Failure of ELLIP1, argument zero "
END IF

END FUNCTION

FUNCTION ELLIP2 (p, PHI1, CA)

'*****************************************************************************
'          A function to compute the Elliptic Integral of the Second Kind _
'*****************************************************************************

'
' The desired accuracy is the square of CA.
'
PI = 3.141592653589793#
CB = .01 * CA ^ 2
QQC = SQR(1 - p ^ 2)
X = TAN(PHI1)
IF X = 0 THEN
   ELLIP2 = 0
```

```
ELSEIF QQC < > 0 THEN
   QC = QQC
   A = 1
   B = QQC ^ 2
   C = X ^ 2
   D = 1 + C
   pp = SQR((1 + QC ^ 2 * C) / D)
   D = X / D
   C = D / (2 * pp)
   Z = A - B
   EYE = A
   A = .5 * (B + A)
   Y = ABS(1 / X)
   F = 0
   L = 0
   EM = 1
   QC = ABS(QC)
4  B = B + EYE * QC
   E = EM * QC
   G = E / pp
   D = D + F * G
   F = C
   EYE = A
   pp = pp + G
   C = .5 * (D / pp + C)
   G = EM
   EM = EM + QC
   A = .5 * (B / EM + A)
   Y = Y - E / Y
   IF Y = 0 THEN Y = SQR(E) * CB
   IF ABS(G - QC) > CA * G THEN
      QC = SQR(E) * 2
      L = L + L
      IF Y < 0 THEN L = L + 1
      GOTO 4
   END IF
   IF Y < 0 THEN L = L + 1
   E = (ATN(EM / Y) + PI * L) * A / EM
   IF X < 0 THEN E = -E
   ELLIP2 = E + C * Z
ELSE
   PRINT "Failure of ELLIP2, argument zero "
END IF

END FUNCTION

SUB ParameterP (p, q, TOL)

'*  Iterative Module used to find the parameter p using the following eqn.
'*              q = K(p) - F[p,ASin(1/p*sqr(2)]
'*  A Bisection Iterative Procedure is Employed

PI = 3.1415926535897793#
pL = 1 / SQR(2)            'Lower bound of p guess --> p< 1/sqr(2) --> invalid ArcSine
pR = .99999999999999999#   'Upper bound of p (p=1 --> infinitely long beam)
CA = .0001#

DO
   pM = (pL + pR) / 2
   t1 = q - CELLIP1(pM, CA) + ELLIP1(pM, ASin(1 / (pM * SQR(2))), CA)
   t2 = q - CELLIP1(pR, CA) + ELLIP1(pR, ASin(1 / (pR * SQR(2))), CA)
```

```
    IF (t1 * t2) > 0 THEN pR = pM ELSE pL = pM
LOOP UNTIL (ABS(t1) <= TOL)
p = pM

END SUB

SUB SlopePHI (s, p, PhiB, Phi, k, TOL)

'*  Iterative Module used to find the slope PHI at any point using the following eqn.  *
'*                    s = [F(p,Phi) - F(p,PhiB)]/k                                      *

PI = 3.1415926535897#
Stp = .1#
FLAG = 0
CA = .0001
PhiL = 0                                'initial guess
PhiR = PI / 2
hh = Phi * .01

DO
    PhiM = (PhiL + PhiR) / 2#
    t1 = k * s - ELLIP1(p, PhiM, CA) + ELLIP1(p, PhiB, CA)
    t2 = k * s - ELLIP1(p, PhiR, CA) + ELLIP1(p, PhiB, CA)
    IF (t1 * t2) > 0 THEN PhiR = PhiM ELSE PhiL = PhiM

LOOP UNTIL (ABS(t1) <= TOL)
Phi = PhiM

END SUB
```

APPENDIX

B

Analytical Solutions of the Cantilever Under Uniform Normal Load

In order to verify the segmental shooting technique, knowns solutions were used. For a horizontal cantilever under uniform normal load, Mitchell's [8] analytical solutions in terms of elliptic integrals of the first and second kind were employed. A bisection root finding procedure was implemented in Microsoft QuickBASIC® to numerically computed this analytical solutions. The following details the source listing.

```
'*****************************************************************************************

    This program is designed to evaluate the final geometry of a horizontal cantilever beam subjected to a vertical point load
    at the end using the analytical solution formulated by Mitchell.

    The program would be required to evaluate Elliptic  Integrals of the First and Second Kind. The Elliptic Integrals were
    computed numerically using an infinite series expansion detailed in "Numerical Recipies".

                                    Version 1.0  Victor Tam  July 6, 1990

'*****************************************************************************************

DECLARE FUNCTION ASin# (A#)
DECLARE FUNCTION CELLIP1# (P#, CA#)
DECLARE FUNCTION ELLIP1# (P#, PHI1#, CA#)
DEFINT I-N
DEFDBL A-H, O-Z

'*** Define Constants

COLOR 15, 1
CLS
Q = 25 ^ 2                                  '*** Applied Load Parameter
CA = .000001                                '*** Sqr(Accuracy) for Elliptic Integrals
CONST PI = 3.1415926535589793#
Dk = (15 * PI / 180)
pL = 0                                      '*** Lower Bound
pR = 10 * PI                                '*** Upper Bound
TOL = 1E-12
Stp = .001                                  '*** Iterative Step Size

DO
   flag = 0
   LOCATE 1, 1: PRINT pM
   pM = (pL + pR) / 2
   IF (pL >= (90 * PI / 180)) THEN
      s! = INT(pL / PI)
      IF s! = 0 THEN s! = 1
      ss = ABS(pL - s! * PI)
      Sign = SGN(pL - s! * PI)
      IF (ss >= (90 * PI / 180)) THEN
         s1! = INT(ss / PI)
         IF s1! = 0 THEN s1! = 1
         ss1 = ABS(ss - s1! * PI)
         Sign1 = SGN(ss - s1! * PI)
         flag = 1
      END IF
      Dk = (15 * PI / 180)
      IF flag = 1 THEN
         t1 = 2 * s! * CELLIP1(Dk, CA) + Sign * (2 * s1! * CELLIP1(Dk, CA) + Sign1 * ELLIP1(Dk, ss1, CA)) - 3 ^ .25 *
Q ^ (1 / 3)
      ELSE
         t1 = 2 * s! * CELLIP1(Dk, CA) + Sign * (ELLIP1(Dk, ss, CA)) - 3 ^ .25 * Q ^ (1 / 3)
      END IF
   ELSE
         t1 = ELLIP1(Dk, pL, CA) - 3 ^ .25 * Q ^ (1 / 3)
   END IF
   IF (pM >= (90 * PI / 180)) THEN
      s! = INT(pM / PI)
      IF s! = 0 THEN s! = 1
      ss = ABS(pM - s! * PI)
```

```
      Sign = SGN(pM - s! * PI)
      IF (ss >= (90 * PI / 180)) THEN
         s1! = INT(ss / PI)
         IF s1! = 0 THEN s1! = 1
         ss1 = ABS(ss - s1! * PI)
         Sign1 = SGN(ss - s1! * PI)
         flag = 1
      END IF
      Dk = (15 * PI / 180)
      IF flag = 1 THEN
         t2 = 2 * s! * CELLIP1(Dk, CA) + Sign * (2 * s1! * CELLIP1(Dk, CA) + Sign1 * ELLIP1(Dk, ss1, CA)) - 3 ^ .25 *
Q ^ (1 / 3)
      ELSE
         t2 = 2 * s! * CELLIP1(Dk, CA) + Sign * (ELLIP1(Dk, ss, CA)) - 3 ^ .25 * Q ^ (1 / 3)
      END IF
   ELSE
         t2 = ELLIP1(Dk, pM, CA) - 3 ^ .25 * Q ^ (1 / 3)
   END IF
   LOCATE 2, 1: PRINT t2
   IF (t1 * t2) > 0 THEN pL = pM ELSE pR = pM

LOOP UNTIL (ABS(t1) <= TOL)
   Omega = pM
   rMAX = 2 / (Q ^ (1 / 3))
   PRINT Omega: SOUND 5000, 1
   rL = SQR((4 * (1 - COS(Omega))) / (Q ^ (2 / 3) * ((SQR(3) - 1) * COS(Omega) + SQR(3) + 1)))
   ThetaL = PI + ASin(Q * rL ^ 3 / 8)
   Gamma = 4 / 3 * ThetaL
   h = rL * COS(ThetaL)
   d = rL * SIN(ThetaL)
   PRINT Gamma
   PRINT rL, h, d

   QQ = INT(Q * 10) / 10
   OutFile$ = "e:\Streu.tmp"
   OPEN OutFile$ FOR OUTPUT AS #1
   FOR Theta = (Gamma) TO 0 STEP -Stp
      Sign1 = SGN(8 * SIN(3 * (Gamma - Theta)) / Q)
      r = ABS((8 * SIN(3 * (Gamma - Theta)) / Q)) ^ (1 / 3)
      IF Sign1 = -1 THEN r = -r
      h = r * COS(Theta)
      d = r * SIN(Theta)
      PRINT #1, CSNG(h), CSNG(d)
   NEXT Theta
END

   FOR Theta = Gamma TO PI / 2 STEP Stp
      Sign1 = SGN(8 * SIN(3 * (Gamma - Theta)) / Q)
      r = ABS((8 * SIN(3 * (Gamma - Theta)) / Q)) ^ (1 / 3)
      IF Sign1 = -1 THEN r = -r
      h = r * COS(Theta)
      d = r * SIN(Theta)
      PRINT #1, CSNG(h), CSNG(d)
   NEXT Theta
   SOUND 3000, 1
END
   FOR r = 0 TO rMAX STEP Stp
      Theta = Gamma - 1 / 3 * ((ASin(Q * r ^ 3 / 8)))
      h = r * COS(Theta) + PI / 2)
      d = r * SIN(Theta) + PI / 2)
      PRINT #1, CSNG(h), CSNG(d), CSNG(Theta), CSNG(r)
```

```
    NEXT r
    FOR r = rMAX - rMAX * .001 TO -rL STEP -Stp
       Theta = Gamma - 1 / 3 * (PI - (ASin(Q * r ^ 3 / 8)))
       h = r * COS(Theta)
       d = r * SIN(Theta)
       PRINT #1, CSNG(h), CSNG(d), CSNG(Theta), CSNG(r)
    NEXT r
    Theta = Gamma - 1 / 3 * (PI - (ASin(Q * rL ^ 3 / 8)))
    h = rL * COS(Theta)
    d = rL * SIN(Theta)
    SOUND 4000, 1

FUNCTION ASin (A)

'*****************************************************************************************************
          A function to compute the ArcSine of a Number in terms of ArcTangent
'*****************************************************************************************************

    ASin = ATN(SQR(A ^ 2 / (1 - A ^ 2)))

END FUNCTION

FUNCTION CELLIP1 (P, CA)

'*****************************************************************************************************
          A function to compute the Complete Elliptic Integral of the First Kind
'*****************************************************************************************************

  QQC = SQR(1 - P ^ 2)
  QC = ABS(QQC)
  A = 1
  B = 1
  PP = 1
  E = QC
  EM = 1
11 F = A
  A = A + B / PP
  G = E / PP
  B = B + F * G
  B = B + B
  PP = G + PP
  G = EM
  EM = QC + EM
  IF ABS(G - QC) > G * CA THEN
     QC = SQR(E)
     QC = QC + QC
     E = QC * EM
     GOTO 11
   END IF
   CELLIP1 = (PI / 2) * (B + A * EM) / (EM * (EM + PP))
END FUNCTION

FUNCTION ELLIP1 (P, PHI1, CA)

'*****************************************************************************************************
          A function to compute the Complete Elliptic Integral of the Second Kind
'*****************************************************************************************************

'
' The desired accuracy is the square of CA.
'
```

```
    CB = .01 * CA ^ 2
    QQC = SQR(1 - P ^ 2)
    X = TAN(PHI1)
    IF X = 0 THEN
       ELLIP1 = 0
    ELSEIF QQC < > 0 THEN
       QC = QQC
       A = 1
       B = 1
       C = X ^ 2
       d = 1 + C
       PP = SQR((1 + QC ^ 2 * C) / d)
       d = X / d
       C = d / (2 * PP)
       Z = A - B
       EYE = A
       A = .5 * (B + A)
       Y = ABS(1 / X)
       F = 0
       L = 0
       EM = 1
       QC = ABS(QC)
1      B = B + EYE * QC
       E = EM * QC
       G = E / PP
       d = d + F * G
       F = C
       EYE = A
       PP = PP + G
       C = .5 * (d / PP + C)
       G = EM
       EM = EM + QC
       A = .5 * (B / EM + A)
       Y = Y - E / Y
       IF Y = 0 THEN Y = SQR(E) * CB
       IF ABS(G - QC) > CA * G THEN
          QC = SQR(E) * 2
          L = L + L
          IF Y < 0 THEN L = L + 1
          GOTO 1
       END IF
       IF Y < 0 THEN L = L + 1
       E = (ATN(EM / Y) + PI * L) * A / EM
       IF X < 0 THEN E = -E
       ELLIP1 = E + C * Z
    ELSE
       PRINT "Failure of ELLIP1, argument zero "
    END IF
END FUNCTION
```

APPENDIX
C

SURFACE PLOT ACCURACY

A commercial package SURFER® by Golden Software® is employed to generate contour plots based on data from the segmental approach. Since the accuracy of such commercial packages is not known, a hemisphere with governing equation

$$X^2+Y^2+Z^2 - 1 \qquad (C.1)$$

is produced numerically for $-1 \leq X \leq 1$, $-1 \leq Y \leq 1$ and $0 \leq Z \leq 1$. A total of 3600 discrete data points were generated in order to form a reasonable representation of the hemisphere. The corresponding contour plot was produced using SURFER® in much the same way the contours in this thesis were constructed. The contours of $Z = 0.5$ and $Z = 1.0$ are shown in Figure C.1. Since interpolations of the given discrete data points are involved, comparing known boundary points of the hemisphere would provide indication of the software's accuracy. The contour of $Z = 0$ was compared at $X = -1$ and 1. It is expected that at $X = -1$ and 1, $Y = 0$. However, as shown in Figure C.1, this is clearly not the case. Furthermore, the contour of $Z = 0$ appears to be "*jagged*" at the borders prompting possible interpolation errors.

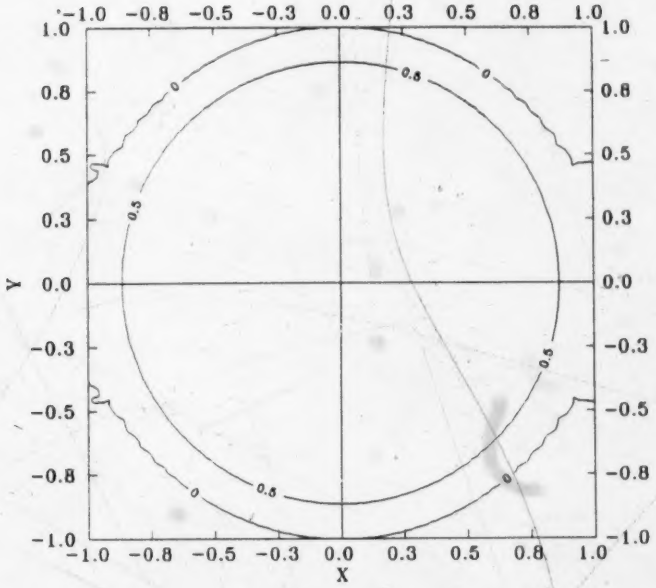


Figure C.1 Contour of a Hemisphere for Z = 0.5 and 1.0 using 100 X 100 Normal Search Grid

A few methods are available to improve the accuracy of the contour plot. These methods are to increase the grid size of the plot and increasing the search radius of the interpolation. Both of these method would directly result in an increase in computing time. The plot shown in Figure C.1 was produced with a 100 X 100 normal grid and a search radius of 10 data points. Another contour, shown in Figure C.2 with a 200 X 200 octant grid and a search radius of 10 data points were made. An octant grid searches the data point within an octant in all directions. This will allow a more accurate interpolation of the function. However, as shown in Figure C.2, the extra effort does not lead to apparent improvements of the plot.

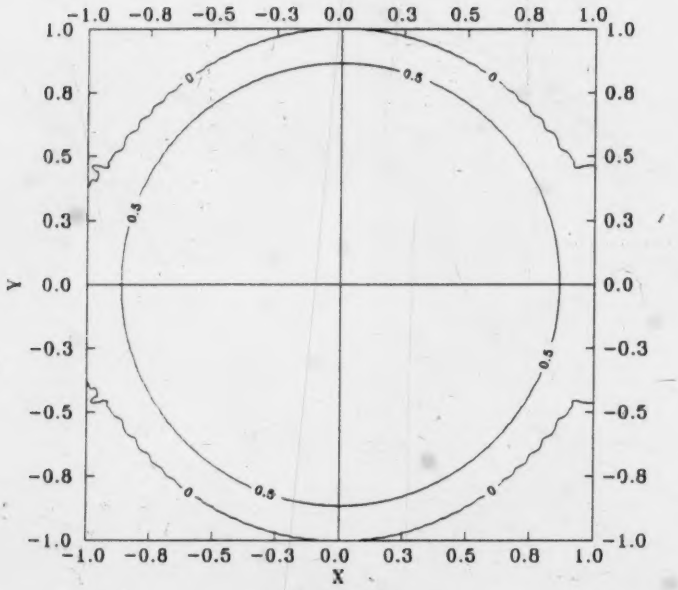


Figure C.2 Contour of a Hemisphere for Z = 0.5 and 1.0 using 200 X 200 Octant Search Grid

APPENDIX
D

SEGMENTAL SHOOTING TECHNIQUE SOURCE LISTING

The segmental shooting technique source code is implemented in MicroSoft QuickBASIC 4.5®. The programming language features modular programming and is employed here. The source listing is divided into various different modules, each of which has a brief explanation of its functions. A substantial portion of the source listing consists of a plotting routine as well as data I/O routines so as to make it more general and robust. The segmental method of solutions can be found in subroutine SolveForcePtLd, SolveForceNORMLD and SolveForceDISTLD. The Newton Raphson shooting procedure is implemented in modules NewRap1, NewRap2 and NewRap3 denoting one, two and three variables iterative procedure.

```
DECLARE SUB SolveForcePtLd (Parameters#(), Delta#, EI#, RadCur#, INumSeg%, TOL#, Energy#,
q#)
DECLARE SUB Screen3 (Length#(), RadCur#(), INumSeg%(), E#(), DInertia#(), TOL#(), StartNode#(),
StartNode$(), IFlagStartNode%(), ISecNum%, INumSections%)
DECLARE SUB PlotGeom (INumSeg%, TOL#)
DECLARE SUB NewRap2 (Z2#(), Parameters#(), SecondKnown1#, SecondKnown2#, INumSeg%(), EI#(),
RadCur#(), Length#(), TOL#(), IFlagStartNode%(), IFlagEndNode%(), StartNode#(), ILoadCond%, q#(),
ISecNum%, INumSections%)
DECLARE SUB NewRap1 (Z1#(), Parameters#(), SecondKnown1#, INumSeg%(), EI#(), RadCur#(),
Length#(), TOL#(), IFlagStartNode%(), IFlagEndNode%(), StartNode#(), ILoadCond%, q#(), ISecNum%,
INumSections%)
DECLARE SUB NewRap3 (Z3#(), Parameters#(), SecondKnown1#, SecondKnown2#, SecondKnown3#,
INumSeg%(), EI#(), RadCur#(), Length#(), TOL#(), IFlagStartNode%(), IFlagEndNode%(), StartNode#(),
ILoadCond%, q#(), ISecNum%, INumSections%)
DECLARE SUB NewRap (Z1#(), Z2#(), Z3#(), Parameters#(), SecondKnown1#, SecondKnown2#,
SecondKnown3#, INumSeg%(), EI#(), RadCur#(), Length#(), TOL#(), IShootDim%, IFlagStartNode%(),
IFlagEndNode%(), StartNode#(), ILoadCond%, q#(), ISecNum%, _
INumSections%)
DECLARE SUB SolveForces (ILoadCond%, Parameters#(), Delta#, EI#, RadCur#, INumSeg%, TOL#,
Energy#, q#)
DECLARE SUB Screen2 (q#(), ILoadCond%, ISecNum%)
DECLARE SUB SolveForceDistLd (Parameters#(), Delta#, EI#, RadCur#, INumSeg%, TOL#, Energy#,
q#)
DECLARE SUB SolveForceNormLd (Parameters#(), Delta#, EI#, RadCur#, INumSeg%, TOL#, Energy#,
q#)
DECLARE SUB NoYes (Answer%, Message$, LocateY%)
DECLARE SUB DefColour ()
DECLARE SUB InputEndKnown (EndNode#(), EndNode$(), IKnown2%, IFlagEndNode%(),
IFlagStartNode%(), IShootDim%, ISecNum%)
DECLARE SUB InitBnd (Z1#(), Z2#(), Z3#(), SecondKnown1#, SecondKnown2#, SecondKnown3#,
IShootDim%, StartNode#(), EndNode#(), IFlagStartNode%(), IFlagEndNode%(), ISecNum%)
DECLARE SUB SelectParStart (i%, Z1#(), Z2#(), Z3#(), IShootDim%, StartNode#(), IFlag2%())
DECLARE SUB Screen1 (Length#(), RadCur#(), INumSeg%(), E#(), DInertia#(), EI#(), TOL#(),
INumSections%, ISecNum%)
DECLARE SUB Gauss (A#(), X#(), B#())
DECLARE SUB FirstNode3 (Parameters#(), ZZ#(), IFlagStartNode%(), StartNode#(), ISecNum%)
DECLARE SUB FirstNode1 (Parameters#(), ZZ#, IFlagStartNode%(), StartNode#(), ISecNum%)
DECLARE SUB Centre (Text$, YCoord%)
DECLARE SUB DefStartNode (StartNode#(), StartNode$(), IFlagStartNode%())
DECLARE SUB InputStartKnown (StartNode#(), StartNode$(), IUnKnown%, IFlagStartNode%())
DECLARE SUB InputStartUnKnown (StartNode#(), StartNode$(), IUnKnown%, IFlagStartNode%())
DECLARE SUB SelectParEnd (i%, SecondKnown1#, SecondKnown2#, SecondKnown3#, IShootDim%,
EndNode#(), IFlag3%())
DECLARE SUB Screen0 (INumSections%)
DECLARE SUB FirstNode2 (Parameters#(), ZZ#(), IFlagStartNode%(), StartNode#(), ISecNum%)
DECLARE SUB Marquee (X$, Row%, Colr%)
DECLARE SUB Scroll (Message$, IRow%, Icolor%)
DECLARE SUB Pause (Ticks%)
DECLARE SUB QPrint (X$, Colr%, Page%)
DECLARE FUNCTION Monitor% ()
DECLARE FUNCTION EGAMem% ()
```

'~
'~ SEGMENTAL SHOOTING TECHNIQUE
'~
'~ UNIVERSTIY OF ALBERTA, EDMONTON ALBERTA
'~
'~ VICTOR TAM
'~
'~

'* SEGMENTAL TECHNIQUE - A numerical method for analysing large deformation nonlinear planar rod problems under various loading. This numerical technique is divided into the segmental approach and the shooting technique.

'* SEGMENTAL APPROACH - This approach involves dividing a largely deformed into numerous small segments. The nonlinear rod problem is then solved segment by segment using a linear approximation on each segment. The segments are assembled together using geometric and force compatibility conditons.

'* SHOOTING TECHNIQUE - The linearization procedure in conjunction with the solving of the linear differntial essentially converts a boundary value problem into an initial value problem. Consequently, a false position shooting procedure is employed to iterative to the desired solutions(s).

Various Sub-Modules are employed to solve large deformation rod problems in much generatlity. Generality involving different curvatures, various loading conditons (concentrated, distributed and normal load), strain energy and total potential energy considerations ... etc. Much efforts have been put in to generate computer codes that automates the problem definition process, solution convergence and data generation. This computer code is neither an illustration of optimized computer programming nor designed to optimally execute the numerical procedure. The codes would require much improvemnts for speed, accuracy and efficiency. The correct execution of this code requires the complete understanding of the underlying theroies pertaining to the segmental shooting technique as well as knowldege regarding the design of this program. Faliure to do so will result in erronous solutions. The author of this computer code is not responsible for any damage direct or indirect, mental or physical induced because of its use. The author reserves the right to make any modifications to the code without notification.

'**********

Sub-Modules

*** GENTEC ***

GENTEC - Main program for solving a GENeral problem under various loading using the Segmental Shooting TEChnique

Center - centers a character string on screen.
DefColour - Maps colour number with corresponding colour name.
DefStartNode - Define initial parameters at the start node of the rod.
FirstNode1 - Assign one initial known value at the start node.

FirstNode2 - Assign two initial known values at the start node.
FirstNode3 - Assign three initial known values at the start node.
Gauss - Gauss Elimination subroutine for solving systems of equations.
InitBnd - Define flags for each initial start node value.
InputEndKnown - Input the known end conditions.
InputStartKnown - Input the known start conditons.
InputStartUnknown - Input the guess for the unknown start conditions.
Marquee - Text Scrolling routine.
NewRap - invoke 1, 2 and 3 varibales Newtwon Raphson shooting procedure.
NewRap1 - invoke 1 variable Newtwon Raphson shooting procedure.
NewRap2 - invoke 2 variables Newtwon Raphson shooting procedure.
NewRap3 - invoke 3 variables Newtwon Raphson shooting procedure.
NoYes - Prompt for Yes or No answer and return corresponding reply.
PlotGeom - Plot geometry of the deflected shape.
Screen0 - First text screen of this program.
Screen1 - Second text Screen of this program.
Screen2 - Third text screen of this program.
Screen3 - Fourth text screen of this program.
Scroll - Scrolling of text.
SelectParEnd - Compare the computed parameter and known parameter at the end node.
SelectParStart - initial a small perturbation of the start guess(es) for Newton Raphson procedure.
SolveForceDistLd - Segmental Approach for solving rods under distributed load.
SolveForceNormLd - Segmental Approach for solving rods under normal load.
SolveForcePtLd - Segmental Approach for solving rods under point load.
SolveForces - Subroutine for invoking different solving routine for different loading conditions.

```
'*****************************************************************************************

DEFINT I-K              '*** Define Interger Variables I to K.
DEFDBL A-H, L-Z         '*** Define Double Precision Variables A to H and L to Z.
OPTION BASE 0           '*** Assign Array to Start at Index 0

'*****************************************************************************************
```

LIST OF VARIABLES USED IN MAIN():

Nomenclature:

Each structure of interset is divided into sections, each section having constant material properties, radius of curvature, number of segments. The individual section of the structure is then divided into numerous segments. Therfore, a structure can be composed of sections with different material properties, numeber of segments, radius of curvature ... etc.

*** ARRAYS
Z1() - guesses for 1 dimension shooting
Z2() - guesse for 2 dimension shooting
Z2() - guessed for 3 dimension shooting
StartNode() - Initial values at the start (X, Y, Gamma, T, V, M)
StartNode$() - Strings (Descriptions) of each element for the initial values
IFlagStartNode() - Flags (ones or zeros) indicating the status of each entry
EndNode() - End values at the start (X, Y, Gamma, T, V, M)
EndNode$() - Strings (Descriptions) of each element for the initia values

```
IFlagEndNode() - Flags (ones or zeros) indicating the status of each entry
Length() - Length of the rod for various sections
RadCur() - Radius of Curvature of the rod for different sections
INumSeg() - Number of segments used for each section
E() - Yound's Modulus for each section
DInertia() - Inertia of each section
EI() - Modulus of Rigidity of each section
TOL() - Numerical Tolerance for each section
Parameters() - a copy of StartNode() for the iterative procedure
q() - Load parameter for each section

VARIABLES:
ISecNum - the total number of sections defined on a structure
Answer - a boolean return to the answer of a prompt (1=Yes, 0=NO)

'*****************************************************************************************


DIM Z1(30), Z2(30, 30), Z3(30, 30), IFlagStartNode(6)
DIM StartNode(6), StartNode$(6), IFlagEndNode(6)
DIM EndNode(6), EndNode$(6), Length(20), RadCur(20), INumSeg(20), E(20)
DIM DInertia(20), EI(20), TOL(20), Parameters(6), q(20)

'       ***************************** Main Program ********************************


  CLEAR , , 2000              '*** Clear All Memory and Set Stack Size
  Answer% = 1
  WHILE Answer%
    CLEAR , , 2000
    DefColour                 '*** Invoke Define Colour Routine
    KEY(7) ON: ON KEY(7) GOSUB BreakOut    '*** Set EXIT Key F7 Trap
    Screen0 INumSections      '*** Invoke First Text Screen

     '***  Loop the Data Input Screens 1 and 2 until all sections defined are exhausted  ***

    FOR ISecNum = 1 TO INumSections STEP 1
      Screen1 Length(), RadCur(), INumSeg(), E(), DInertia(), EI(), TOL(), INumSections, ISecNum
      Screen2 q(), ILoadCond, ISecNum
    IF ISecNum = 1 THEN
      DefStartNode StartNode(), StartNode$(), IFlagStartNode()
      InputStartKnown StartNode(), StartNode$(), IUnKnown, IFlagStartNode()
      InputStartUnKnown StartNode(), StartNode$(), IUnKnown, IFlagStartNode()
    END IF
    NEXT ISecNum

    ISecNum = 1

              '***  Input and Initialize the Initial Known Boundary Conditions  ***

    InputEndKnown EndNode(), EndNode$(), IKnown2, IFlagEndNode(), IFlagStartNode(), IShootDim,
```

```
ISecNum
      InitBnd Z1(), Z2(), Z3(), SecondKnown1, SecondKnown2, SecondKnown3, IShootDim, StartNode(),
EndNode(), IFlagStartNode(), IFlagEndNode(), ISecNum
      Screen3 Length(), RadCur(), INumSeg(), E(), DInertia(), TOL(), StartNode(), StartNode$(),
IFlagStartNode(), ISecNum, INumSections

         '*** Invoke Newton Raphson Routine based on the number of initial unknowns ***
         '*** Within the Newton Raphson Procedure, the segmental approach is called ***

      NewRap Z1(), Z2(), Z3(), Parameters(), SecondKnown1, SecondKnown2, SecondKnown3,
INumSeg(), EI(), RadCur(), Length(), TOL(), IShootDim, IFlagStartNode(), IFlagEndNode(), StartNode(),
ILoadCond, q(), ISecNum, INumSections

'*** Exit Segmental's Approach and prompts for another run ***

      NoYes Answer%, "ReStart?", 15
   WEND

      '*** Terminate Program Execution and Close all File once Exit Key (F7) is trapped ***

BreakOut:
   CLOSE
   END
RETURN

END

   '******************************** END MAIN **********************************


'*****************************************************************************************
'* CENTRE     -     Given a string Text$ and vertical position YCoord, the text Text$ is
                    automatically centered horizontally on vertical the vertical position YCoord

'*****************************************************************************************

SUB Centre (Text$, YCoord%)

      TextLength = LEN(Text$)
      XCoord% = (80 - TextLength) \ 2
      LOCATE YCoord%, XCoord%
      PRINT Text$;

END SUB


'*****************************************************************************************
'* DefColour - Define the colour on screen according to the corresponding names.
'*****************************************************************************************
```

```
SUB DefColour

SHARED IBlack, IBlue, IGreen, ICyan, IRed, IMagenta, IBrown, IWhite, IDarkGrey, ILightBlue,
ILightCyan, ILightRed, ILightMagenta, ILightYellow, IBrWhite, IBlink AS INTEGER

DIM Key$(10)

  Key$(1) = " HELP "

  ''M=Monitor%            'get the monitor type
  'Memory = EGAMem% * 64   'if it's an EGA, get its memory
  ''call clock(1,1,15,6)

  SELECT CASE M
     CASE 1
        'Monochrome adapter"
        SCREEN 0
     CASE 2
        '"Hercules card"
        SCREEN 0
     CASE 3
        '"CGA adapter or an EGA emulating a CGA"
        'SCREEN 2
     CASE 4
        'EGA card with a monochrome monitor,"
        SCREEN 10
     CASE 5
        'EGA card with a COLOR monitor,"
        'SCREEN 9
     CASE 6
        'VGA adapter with a monochrome monitor"
        SCREEN 10
     CASE 7
        'VGA adapter with a COLOR monitor"
        SCREEN 11
  END SELECT

  IBlack = 0
  IBlue = 1
  IGreen = 2
  ICyan = 3
  IRed = 4
  IMagenta = 5
  IBrown = 6
  IWhite = 7
  IDarkGrey = 8
  ILightBlue = 9
  ILightGreen = 10
  ILightCyan = 11
  ILightRed = 12
  ILightMagenta = 13
```

```
    lLightYellow = 14
    lBrWhite = 15
    lBlink = 16

    'KEY ON
    'COLOR 1, 15: KEY 1, Key$(1)

END SUB

'*****************************************************************************************
'* DefStartNode - Initialize the Initial values array and assign a text description to each initial value
'*
'*                    StartNode(1) = X1 value
'*                    StartNode(2) = Y1 value
'*                    StartNode(3) = Gamma value
'*                    StartNode(4) = M1 value
'*                    StartNode(5) = V1 value
'*                    StartNode(6) = T1 value
'*
'*****************************************************************************************

SUB DefStartNode (StartNode(), StartNode$(), lFlagStartNode())

    StartNode$(1) = "X1"
    StartNode$(2) = "Y1"
    StartNode$(3) = "Gamma1"
    StartNode$(4) = "M1"
    StartNode$(5) = "V1"
    StartNode$(6) = "T1"

    FOR i = 1 TO 6
      StartNode(i) = 0#
      lFlagStartNode(i) = 0
    NEXT i

END SUB

'*****************************************************************************************
'* FirstNode1 - Assign initial values to the StartNode array for 1-D shooting
'*****************************************************************************************

SUB FirstNode1 (Parameters(), ZZ, lFlagStartNode(), StartNode(), lSecNum)

    'StartNode(1) = X1 value:    Parameters(1)
    'StartNode(2) = Y1 value:    Parameters(2)
```

```
'StartNode(3) = Gamma value;   Parameters(3)
'StartNode(4) = M1 value;      Parameters(4)
'StartNode(5) = V1 value;      Parameters(5)
'StartNode(6) = T1 value;      Parameters(6)

ISecNum = 1
FOR i = 1 TO 6

   IF IFlagStartNode(i) = 1 THEN
      Parameters(i) = StartNode(i)
   END IF
   IF IFlagStartNode(i) = 2 THEN
      Parameters(i) = ZZ
   END IF

NEXT i


'XGlobal = 0#
'YGlobal = 0#
'M1 = 0#
'Gamma = ZZ,     'Guess Slope
'T1 = -(P * COS(Gamma))
'V1 = -(P * SIN(Gamma))

END SUB

'*****************************************************************************************
'*   FirstNode2 <-- Assign initial values to the StartNode array for 2-D shooting
'*****************************************************************************************

SUB FirstNode2 (Parameters(), ZZ(), IFlagStartNode(), StartNode(), ISecNum)


'StartNode(1) = X1 value;      Parameters(1)
'StartNode(2) = Y1 value;      Parameters(2)
'StartNode(3) = Gamma value;   Parameters(3)
'StartNode(4) = M1 value;      Parameters(4)
'StartNode(5) = V1 value;      Parameters(5)
'StartNode(6) = T1 value;      Parameters(6)

IFlag2 = 0
ISecNum = 1
FOR i = 1 TO 6

   IF IFlagStartNode(i) = 1 THEN
      Parameters(i) = StartNode(i)
   END IF
   IF IFlagStartNode(i) = 2 AND IFlag2 = 0 THEN
         Parameters(i) = ZZ(1)
```

```
            IFlag2 = IFlag2 + 1
        ELSE
            IF IFlagStartNode(i) = 2 AND IFlag2 = 1 THEN
                Parameters(i) = ZZ(2)
                IFlag2 = IFlag2 + 1
            END IF
        END IF

    NEXT i


    'XGlobal = 0#
    'YGlobal = 0#
    'M1 = 0#
    'V1 = ZZ(1)      'Shoot for Tension
    'Gamma = ZZ(2)   'Shoot for Slope
    'dummy1 = P ^ 2 - V1 ^ 2
    'IF dummy1 >= 0 THEN T1 = -SQR(P ^ 2 - V1 ^ 2) ELSE T1 = SQR(ABS(P ^ 2 - V1 ^ 2))
    'T1 = -SQR(P ^ 2 - V1 ^ 2)

END SUB

'*****************************************************************************************
'* FirstNode3 - Assign initial values to the StartNode array for 3-D shooting
'*****************************************************************************************

SUB FirstNode3 (Parameters(), ZZ(), IFlagStartNode(), StartNode(), ISecNum)

    'StartNode(1) = X1 value;      Parameters(1)
    'StartNode(2) = Y1 value;      Parameters(2)
    'StartNode(3) = Gamma value;   Parameters(3)
    'StartNode(4) = M1 value;      Parameters(4)
    'StartNode(5) = V1 value;      Parameters(5)
    'StartNode(6) = T1 value;      Parameters(6)

    IFlag3 = 0
    ISecNum = 1
    FOR i = 1 TO 6

        IF IFlagStartNode(i) = 1 THEN
            Parameters(i) = StartNode(i)
        END IF
        IF IFlagStartNode(i) = 2 AND IFlag3 = 0 THEN
            Parameters(i) = ZZ(1)
            IFlag3 = IFlag3 + 1
        ELSE
            IF IFlagStartNode(i) = 2 AND IFlag3 = 1 THEN
                Parameters(i) = ZZ(2)
                IFlag3 = IFlag3 + 1
```

```
            ELSE
               IF IFlagStartNode(i) = 2 AND IFlag3 = 2 THEN
                  Parameters(i) = ZZ(3)
                  IFlag3 = IFlag3 + 1
               END IF
            END IF
         END IF

      NEXT i

      'XGlobal = 0#
      'YGlobal = 0#
      'M1 = 0#
      'V1 = ZZ(1)      'Shoot for Tension
      'Gamma = ZZ(2)   'Shoot for Slope
      'Victor = ZZ(3)

      'dummy1 = P ^ 2 - V1 ^ 2
      'IF dummy1 >= 0 THEN T1 = -SQR(P ^ 2 - V1 ^ 2) ELSE T1 = SQR(ABS(P ^ 2 - V1 ^ 2))
      'T1 = -SQR(P ^ 2 - V1 ^ 2)


   END SUB

   SUB Gauss (A(), X(), B())

   '*****************************************************************************************
   '*
   '*       GAUSS is a matrix solver routine that returns the values of the unknowns in the B() vector.
   '*
   '*****************************************************************************************

      Row = UBOUND(A, 1)
      COL = UBOUND(A, 2) + 1
      LAST = Row - 1
      DIM MATRIX(Row, COL)
      FOR i = 1 TO Row
        FOR j = 1 TO (COL - 1)
          MATRIX(i, j) = A(i, j)
        NEXT j
        MATRIX(i, COL) = X(i)
      NEXT i

   '****
   '*                Start overall loop for (ROW-1) pivots
   '****

      FOR i = 1 TO LAST

   '****
   '*                Find the largest remaining term in the I-th column for pivot
```

```
'****

      BIG = 0
      FOR k = i TO Row
        TERM = ABS(MATRIX(k, i))
        IF (TERM - BIG) < 0 OR (TERM - BIG) = 0 THEN 6510 ELSE 61000
61000     BIG = TERM
        l = k
6510    NEXT k

'****
'*                  Check whether a non-zero term has been found
'****

      IF BIG < 0 OR BIG > 0 THEN 6530 ELSE 6520
6520    GOTO 6580

'****
'*                  L-th row has the biggest term -- is I = L
'****

6530    IF (i - l) < 0 OR (i - l) > 0 THEN 6540 ELSE 6550

'****
'*                  I is not equal to L so switch rows I and L
'****

6540    FOR j = 1 TO COL
        TEMP = MATRIX(i, j)
        MATRIX(i, j) = MATRIX(l, j)
        MATRIX(l, j) = TEMP
      NEXT j

'****
'*                  Now start pivotal reduction
'****

6550    PIVOT = MATRIX(i, i)
      NEXTR = i + 1

'****
'*                  For each of the rows after the I-th
'****

      FOR j = NEXTR TO Row

'****
'*                  Multiplying constant for the J-th row is...
'****

        CONNST = MATRIX(j, i) / PIVOT
```

```
'****
'*                  Now reduce each term of the J=th row
'****

        FOR k = i TO COL
          MATRIX(j, k) = MATRIX(j, k) - CONNST * MATRIX(i, k)
        NEXT k
      NEXT j
    NEXT i

'****
'*                  End of pivotal reduction,
'*                  now perform back substitution
'****

    FOR i = 1 TO Row

'****
'*                  IREV is the backward index, going from ROW back to 1
'****

      IREV = Row + 1 - i

'****
'*                  Get Y(IREV) in preparation
'****

      Y = MATRIX(IREV, COL)
      IF (IREV - Row) < 0 OR (IREV - Row) > 0 THEN 6560 ELSE 6570

'****
'*                  Not working on last row, I is 2 or greater
'****

6560  FOR j = 2 TO i

'****
'*                  Work backwards for B(COL), B(COL-1)---,
'*                  substituting previously found values
'****

        k = COL + 1 - j
        Y = Y - MATRIX(IREV, k) * B(k)
      NEXT j

'****
'*                  Finally compute B(IREV)
'****

6570  B(IREV) = Y / MATRIX(IREV, IREV)
    NEXT i
```

```
6580 EXIT SUB

END SUB

'*****************************************************************************************
'* InitBnd - Initialize the start and end boundary conditions and assign the guess(es) to the corresponding parameters
'*****************************************************************************************

SUB InitBnd (Z1(), Z2(), Z3(), SecondKnown1, SecondKnown2, SecondKnown3, IShootDim, StartNode(), EndNode(), IFlagStartNode(), IFlagEndNode(), ISecNum)

DIM IFlag2(3), IFlag3(3)

   FOR i = 1 TO 3
      IFlag2(i) = 0
      IFlag3(i) = 0
   NEXT i

   FOR i = 1 TO 6
      IF IFlagStartNode(i) = 2 THEN SelectParStart i, Z1(), Z2(), Z3(), IShootDim, StartNode(), IFlag2()
      IF IFlagEndNode(i) = 1 THEN SelectParEnd i, SecondKnown1, SecondKnown2, SecondKnown3, IShootDim, EndNode(), IFlag3()
   NEXT i

END SUB

'*****************************************************************************************
'* InputEndKnown - Input screen for the knowns at the end node
'*****************************************************************************************

SUB InputEndKnown (EndNode(), EndNode$(), IKnown2, IFlagEndNode(), IFlagStartNode(), IShootDim, ISecNum)

DIM MenuEnd$(7)
SHARED IBlack, IBlue, IGreen, ICyan, IRed, IMagenta, IBrown, IWhite, IDarkGrey, ILightBlue, ILightCyan, ILightRed, ILightMagenta, ILightYellow, IBrWhite, IBlink AS INTEGER

   CLS
   IFlag1 = 1
   IShootDim = 0
   IKnown2 = 0
   Entry$ = ""
   StartRow = 6
   Centre "Select Item(s) and Please Input End Node Parameters", 3
   FOR i = 1 TO 6
```

```
   IF IFlagStartNode(i) = 2 THEN IShootDim = IShootDim + 1
NEXT i
LOCATE 4, 12
PRINT "At Least"; IShootDim; " Distinct Parameters be Known at the EndNode!"

MenuEnd$(1) = "1)    End Node X Coordinate"
MenuEnd$(2) = "2)    End Node Y Coordinate"
MenuEnd$(3) = "3)    End Node Slope"
MenuEnd$(4) = "4)    End Node Moment Value"
MenuEnd$(5) = "5)    End Node Shear Value"
MenuEnd$(6) = "6)    End Node Tension Value"
MenuEnd$(7) = "7)    Return to Previous Menu"

FOR i = 1 TO 6
   LOCATE StartRow + 2 * (i - 1), 20
   PRINT MenuEnd$(i)
NEXT i
PRINT
PRINT "Select Item:";

WHILE (IFlag1 OR (IKnown2 < > IShootDim))

   WHILE (Entry$ = "")
      Entry$ = INKEY$
   WEND

   SELECT CASE Entry$
      CASE "1"
         IF IFlagEndNode(1) = 0 THEN
            PRINT
            INPUT "Please Enter End Node X Coordinate"; EndNode(1)
            IKnown2 = IKnown2 + 1
            IFlagEndNode(1) = 1
            COLOR IGreen
            LOCATE StartRow, 20
            PRINT MenuEnd$(1)
            COLOR IBrWhite
         END IF
      CASE "2"
         IF IFlagEndNode(2) = 0 THEN
            PRINT
            INPUT "Please Enter End Node Y Coordinate"; EndNode(2)
            IKnown2 = IKnown2 + 1
            IFlagEndNode(2) = 1
            COLOR IGreen
            LOCATE StartRow + 2 * (1), 20
            PRINT MenuEnd$(2)
            COLOR IBrWhite
         END IF
      CASE "3"
         IF IFlagEndNode(3) = 0 THEN
```

```
            PRINT
            INPUT "Please Enter End Node Slope"; EndNode(3)
            IKnown2 = IKnown2 + 1
            IFlagEndNode(3) = 1
            COLOR IGreen
            LOCATE StartRow + 2 * (2), 20
            PRINT MenuEnd$(3)
            COLOR IBrWhite
         END IF
      CASE "4"
         IF IFlagEndNode(4) = 0 THEN
            PRINT
            INPUT "Please Enter End Node Moment Value"; EndNode(4)
            IKnown2 = IKnown2 + 1
            IFlagEndNode(4) = 1
            COLOR IGreen
            LOCATE StartRow + 2 * 3, 20
            PRINT MenuEnd$(4)
            COLOR IBrWhite
         END IF
      CASE "5"
         IF IFlagEndNode(5) = 0 THEN
            PRINT
            INPUT "Please Enter End Node Shear Value"; EndNode(5)
            IKnown2 = IKnown2 + 1
            IFlagEndNode(5) = 1
            COLOR IGreen
            LOCATE StartRow + 2 * 4, 20
            PRINT MenuEnd$(5)
            COLOR IBrWhite
         END IF
      CASE "6"
         IF IFlagEndNode(6) = 0 THEN
            PRINT
            INPUT "Please Enter End Node Tension Value"; EndNode(6)
            IKnown2 = IKnown2 + 1
            IFlagEndNode(6) = 1
            COLOR IGreen
            LOCATE StartRow + 2 * 5, 20
            PRINT MenuEnd$(6)
            COLOR IBrWhite
         END IF
      'case 7
      CASE ELSE
         SOUND 6000, 1
         PRINT
         Centre "Invalid Entry, Please Re-enter", 20
         IF IKnown2 >= IShootDim THEN IFlag1 = 0 ELSE IFlag1 = 1
         END SELECT

   Entry$ = ""
```

```
        LOCATE StartRow + 2 * (i - 1), 1
        PRINT "Select Item:";
        LOCATE StartRow + 2 * (i - 1) + 1, 1
        PRINT "                                        ";
        LOCATE StartRow + 2 * (i - 1), 1
        PRINT "Select Item:";
        IF IKnown2 >= IShootDim THEN IFlag1 = 0

        WEND

END SUB

'*****************************************************************************************
'* InputStartKnown - Input screen for the knowns at the start node
'*****************************************************************************************

SUB InputStartKnown (StartNode(), StartNode$(), IUnKnown, IFlagStartNode())

DIM MenuStart$(7)
SHARED IBlack, IBlue, IGreen, ICyan, IRed, IMagenta, IBrown, IWhite, IDarkGrey, ILightBlue,
ILightCyan, ILightRed, ILightMagenta, ILightYellow, IBrWhite, IBlink AS INTEGER

    CLS
    IFlag1 = 1
    IUnKnown = 0
    Entry$ = ""
    StartRow = 6
    Centre "Select Item(s) and Please Input Start Node Parameters", 3
    Centre "At Least 3 Distinct Parameters be Known!", 4

    MenuStart$(1) = "1)    Start Node X Coordinate"
    MenuStart$(2) = "2)    Start Node Y Coordinate"
    MenuStart$(3) = "3)    Start Node Slope"
    MenuStart$(4) = "4)    Start Node Moment Value"
    MenuStart$(5) = "5)    Start Node Shear Value"
    MenuStart$(6) = "6)    Start Node Tension Value"
    MenuStart$(7) = "7)    Exit Input Known Start Node Parameters"

    'StartNode(1) = X1 value
    'StartNode(2) = Y1 value
    'StartNode(3) = Gamma value
    'StartNode(4) = M1 value
    'StartNode(5) = V1 value
    'StartNode(6) = T1 value

    FOR i = 1 TO 7
        LOCATE StartRow + 2 * (i - 1), 20
        PRINT MenuStart$(i)
    NEXT i
```

```
    PRINT
    PRINT "Select Item:";

    WHILE (IFlag1)

      WHILE (Entry$ = "")
        Entry$ = INKEY$
      WEND

      SELECT CASE Entry$
        CASE "1"
          IF IFlagStartNode(1) = 0 THEN
            PRINT
            INPUT "Please Enter Start Node X Coordinate"; StartNode(1)
            IUnKnown = IUnKnown + 1
            IFlag OUTPUT AS #4
      FOR ISecNum = 1 TO INumSections STEP 1
        SolveForces ILoadCond, Parameters(), Delta(ISecNum), EI(ISecNum), RadCur(ISecNum),
INumSeg(ISecNum), TOL(ISecNum), Energy, q(ISecNum)
      NEXT ISecNum
      CLOSE #1, #4

      FOR j = 1 TO 6
        IF IFlagEndNode(j) = 1 THEN
          SELECT CASE IFlag3
            CASE 0
              F1F101 = Parameters(j) - SecondKnown1
              IFlag3 = IFlag3 + 1
            CASE 1
              F2F101 = Parameters(j) - SecondKnown2
              IFlag3 = IFlag3 + 1
            CASE 2
              F3F101 = Parameters(j) - SecondKnown3
              IFlag3 = IFlag3 + 1
          END SELECT
        END IF
      NEXT j

      IFlag3 = 0

'********************************** Third Pass ********************************

      ZZ(1) = Z3(i - 1, 1)
      ZZ(2) = Z3(i, 2)
      ZZ(3) = Z3(i, 3)
      LOCATE 2, 54
      PRINT "Third Pass . . . "
      FirstNode3 Parameters(), ZZ(), IFlagStartNode(), StartNode(), ISecNum
      OPEN "d:\~Geom.Tmp" FOR OUTPUT AS #1
      OPEN "d:\curve.dat" FOR OUTPUT AS #4
      FOR ISecNum = 1 TO INumSections STEP 1
```

```
        SolveForces ILoadCond, Parameters(), Delta(ISecNum), EI(ISecNum), RadCur(ISecNum),
INumSeg(ISecNum), TOL(ISecNum), Energy, q(ISecNum)
    NEXT ISecNum
    CLOSE #1, #4

    FOR j = 1 TO 6
      IF IFlagEndNode(j) = 1 THEN
        SELECT CASE IFlag3
          CASE 0
            F1F011 = Parameters(j) - SecondKnown1
            IFlag3 = IFlag3 + 1
          CASE 1
            F2F011 = Parameters(j) - SecondKnown2
            IFlag3 = IFlag3 + 1
          CASE 2
            F3F011 = Parameters(j) - SecondKnown3
            IFlag3 = IFlag3 + 1
        END SELECT
      END IF
    NEXT j

    IFlag3 = 0

'********************************* Final Pass ******************************

    ZZ(1) = Z3(i, 1)
    ZZ(2) = Z3(i, 2)
    ZZ(3) = Z3(i, 3)
    LOCATE 2, 54
    PRINT "Final Pass » » » »"
    FirstNode3 Parameters(), ZZ(), IFlagStartNode(), StartNode(), ISecNum
    OPEN "d:\~Geom.Tmp" FOR OUTPUT AS #1
    OPEN "d:\curve.dat" FOR OUTPUT AS #4
    FOR ISecNum = 1 TO INumSections STEP 1
        SolveForces ILoadCond, Parameters(), Delta(ISecNum), EI(ISecNum), RadCur(ISecNum),
INumSeg(ISecNum), TOL(ISecNum), Energy, q(ISecNum)
    NEXT ISecNum
    CLOSE #1, #4

    FOR j = 1 TO 6
      IF IFlagEndNode(j) = 1 THEN
        SELECT CASE IFlag3
          CASE 0
            F1F111 = Parameters(j) - SecondKnown1
            IFlag3 = IFlag3 + 1
          CASE 1
            F2F111 = Parameters(j) - SecondKnown2
            IFlag3 = IFlag3 + 1
          CASE 2
            F3F111 = Parameters(j) - SecondKnown3
            IFlag3 = IFlag3 + 1
```

```
            END SELECT
         END IF
      NEXT j

      IFlag3 = 0

      IF (ABS(F1F111) < TOL(1)) AND (ABS(F2F111) < TOL(1)) AND (ABS(F3F111) < TOL(1))
THEN
         FirstNode3 Parameters(), ZZ(), IFlagStartNode(), StartNode(), ISecNum
         OPEN "d:\~Geom.Tmp" FOR OUTPUT AS #1
         FOR ISecNum = 1 TO INumSections STEP 1
            SolveForces ILoadCond, Parameters(), Delta(ISecNum), EI(ISecNum), RadCur(ISecNum),
INumSeg(ISecNum), TOL(ISecNum), Energy, q(ISecNum)
         NEXT ISecNum
         CLOSE #1,#4
      END IF

      Iteration$ = "Iteration #" + STR$(Iteration#) + " Completed"
      Centre Iteration$, 16: PRINT
      PRINT "Moment 2 = ";
      PRINT USING "+#.##############^^^^^"; Parameters(4);
      PRINT ,
      PRINT "Tension 2 = ";
      PRINT USING "+#.##############^^^^^"; Parameters(6)
      PRINT "Slope 2 = ";
      PRINT USING "+#.##############^^^^^"; Parameters(3);
      PRINT ,
      PRINT "Shear 2 = ";
      PRINT USING "+#.##############^^^^^"; Parameters(5)
      PRINT "X2 = ";
      PRINT USING "+#.##############^^^^^"; Parameters(1);
      PRINT , ,
      PRINT "Y2 = ";
      PRINT USING "+#.##############^^^^^"; Parameters(2)
      FOR j = 1 TO 6
         IF IFlagStartNode(j) = 2 THEN
            SELECT CASE IFlag3
               CASE 0
                  PRINT StartNode$(j); " = ";
                  PRINT USING "+#.###########^^^^^"; ZZ(1)
                  'PRINT ,
                  IFlag3 = IFlag3 + 1
               CASE 1
                  PRINT StartNode$(j); " = ";
                  PRINT USING "+#.###########^^^^^"; ZZ(2)
                  'PRINT ,
                  IFlag3 = IFlag3 + 1
               CASE 2
                  PRINT StartNode$(j); " = ";
                  PRINT USING "+#.###########^^^^^"; ZZ(3)
                  'PRINT ,
```

```
                IFlag3 = IFlag3 + 1
            END SELECT
        END IF
    NEXT j
    IFlag3 = 0
    PRINT
    LOCATE 4, 45
    PRINT "Accuracy 1 = ";
    PRINT USING "+#.##############^^^^"; F1F111
    LOCATE , 45
    PRINT "Accuracy 2 = ";
    PRINT USING "+#.##############^^^^"; F2F111
    LOCATE , 45
    PRINT "Accuracy 3 = ";
    PRINT USING "+#.##############^^^^"; F3F111

    Iteration$ = "Iteration #" + STR$(Iteration#) + " Completed"
    Centre Iteration$, 16: PRINT

    DJacobian(1, 1) = (F1F111 - F1F011) / (Z3(i, 1) - Z3(i - 1, 1))
    DJacobian(1, 2) = (F1F111 - F1F101) / (Z3(i, 2) - Z3(i - 1, 2))
    DJacobian(1, 3) = (F1F111 - F1F110) / (Z3(i, 3) - Z3(i - 1, 3))

    DJacobian(2, 1) = (F2F111 - F2F011) / (Z3(i, 1) - Z3(i - 1, 1))
    DJacobian(2, 2) = (F2F111 - F2F101) / (Z3(i, 2) - Z3(i - 1, 2))
    DJacobian(2, 3) = (F2F111 - F2F110) / (Z3(i, 3) - Z3(i - 1, 3))

    DJacobian(3, 1) = (F3F111 - F3F011) / (Z3(i, 1) - Z3(i - 1, 1))
    DJacobian(3, 2) = (F3F111 - F3F101) / (Z3(i, 2) - Z3(i - 1, 2))
    DJacobian(3, 3) = (F3F111 - F3F110) / (Z3(i, 3) - Z3(i - 1, 3))

    Func(1) = -1# * F1F111
    Func(2) = -1# * F2F111
    Func(3) = -1# * F3F111

    Gauss DJacobian(), Func(), Diff()

    Z3(i + 1, 1) = Z3(i, 1) + Diff(1) * WeightFactor1
    Z3(i + 1, 2) = Z3(i, 2) + Diff(2) * WeightFactor2
    Z3(i + 1, 3) = Z3(i, 3) + Diff(3) * WeightFactor3

    i = i + 1
    SOUND 6000, 1

LOOP UNTIL (ABS(F1F111) < TOL(1)) AND (ABS(F2F111) < TOL(1)) AND (ABS(F3F111) <
TOL(1))

    FirstNode3 Parameters(), ZZ(), IFlagStartNode(), StartNode(), ISecNum
    OPEN "d:\~Geom.Tmp" FOR OUTPUT AS #1
    OPEN "d:\curve.dat" FOR OUTPUT AS #4
    FOR ISecNum = 1 TO INumSections STEP 1
```

```
        SolveForces ILoadCond, Parameters(), Delta(ISecNum), EI(ISecNum), RadCur(ISecNum),
INumSeg(ISecNum), TOL(ISecNum), Energy, q(ISecNum)
    NEXT ISecNum
    CLOSE #1, #4

    LOCATE 17, 1
    PRINT "Moment 2 = ";
    PRINT USING "+#.##############^^^^"; Parameters(4);
    PRINT ,
    PRINT "Tension 2 = ";
    PRINT USING "+#.##############^^^^"; Parameters(6)
    PRINT "Slope 2 = ";
    PRINT USING "+#.##############^^^^"; Parameters(3);
    PRINT ,
    PRINT "Shear 2 = ";
    PRINT USING "+#.##############^^^^"; Parameters(5)
    PRINT "X2 = ";
    PRINT USING "+#.##############^^^^"; Parameters(1);
    PRINT , ,
    PRINT "Y2 = ";
    PRINT USING "+#.##############^^^^"; Parameters(2)
    FOR j = 1 TO 6
      IF IFlagStartNode(j) = 2 THEN
        SELECT CASE IFlag3
          CASE 0
            PRINT StartNode$(j); " = ";
            PRINT USING "+#.############^^^^"; ZZ(1)
            'PRINT ,
            IFlag3 = IFlag3 + 1
          CASE 1
            PRINT StartNode$(j); " = ";
            PRINT USING "+#.############^^^^"; ZZ(2)
            'PRINT ,
            IFlag3 = IFlag3 + 1
          CASE 2
            PRINT StartNode$(j); " = ";
            PRINT USING "+#.############^^^^"; ZZ(3)
            'PRINT ,
            IFlag3 = IFlag3 + 1
        END SELECT
      END IF
    NEXT j
    IFlag3 = 0
    PRINT
    LOCATE 4, 45
    PRINT "Accuracy 1 = ";
    PRINT USING "+#.##############^^^^"; F1F111
    LOCATE , 45
    PRINT "Accuracy 2 = ";
    PRINT USING "+#.##############^^^^"; F2F111
    LOCATE , 45
```

```
    PRINT "Accuracy 3 = ";
    PRINT USING "+#.#############^^^^"; F3F111
     OPEN "d:\segtec.out" FOR output AS #3
    PRINT #3, "X2 = ";
    PRINT #3, USING "+#.#############^^^^"; Parameters(1)
    PRINT #3, "Y2 = ";
    PRINT #3, USING "+#.#############^^^^"; Parameters(2)
    PRINT #3, "Slope 2 = ";
    PRINT #3, USING "+#.#############^^^^"; Parameters(3)
    PRINT #3, "Moment 2 = ";
    PRINT #3, USING "+#.#############^^^^"; Parameters(4)
    PRINT #3, "Shear 2 = ";
    PRINT #3, USING "+#.#############^^^^"; Parameters(5)
    PRINT #3, "Tension 2 = ";
    PRINT #3, USING "+#.#############^^^^"; Parameters(6)
    FOR j = 1 TO 6
       IF IFlagStartNode(j) = 2 THEN
          SELECT CASE IFlag3
             CASE 0
                PRINT #3, StartNode$(j); " = ";
                PRINT #3, USING "+#.##########^^^^"; ZZ(1)
                'PRINT ,
                IFlag3 = IFlag3 + 1
             CASE 1
                PRINT #3, StartNode$(j); " = ";
                PRINT #3, USING "+#.##########^^^^"; ZZ(2)
                'PRINT ,
                IFlag3 = IFlag3 + 1
             CASE 2
                PRINT #3, StartNode$(j); " = ";
                PRINT #3, USING "+#.##########^^^^"; ZZ(3)
                'PRINT ,
                IFlag3 = IFlag3 + 1
             END SELECT
       END IF
    NEXT j
    IFlag3 = 0
    PRINT #3, "Accuracy 1 = ";
    PRINT #3, USING "+#.#############^^^^"; F1F111
    PRINT #3, "Accuracy 2 = ";
    PRINT #3, USING "+#.#############^^^^"; F2F111
    PRINT #3, "Accuracy 3 = ";
    PRINT #3, USING "+#.#############^^^^"; F3F111
  'call clock(1, 1, 14, 0)
    SOUND 4000, 1
    LOCATE 24, 1: PRINT #3, TimeIdx$
    SLEEP
  CLOSE
  PlotGeom INumSeg, TOL(1)
```

```
END SUB

'*****************************************************************************
'* NoYes - prompt for answer (Answer%) to a question (Message$) at location LocateY%
'*****************************************************************************

SUB NoYes (Answer%, Message$, LocateY%)

   Centre Message$, LocateY%
   A$ = INKEY$

   Answer% = -999
   WHILE (Answer% < 0)
      A$ = INKEY$
      WHILE (A$ = "")
         A$ = INKEY$
      WEND
         SELECT CASE A$
            CASE "Y"
               Answer% = 1
            CASE "y"
               Answer% = 1
            CASE "N"
               Answer% = 0
            CASE "n"
               Answer% = 0
            CASE ""
               'do nothing
               Answer% = -999
            CASE ELSE
               Centre "InValid Entry, Please Re-Enter", LocateY% + 1
               Answer% = -999
         END SELECT
   WEND

END SUB

'*****************************************************************************
'* PlotGeom - plot geometry of the deflected shape on screen
'*****************************************************************************

SUB PlotGeom (INumSeg, TOL)

SHARED IBlack, IBlue, IGreen, ICyan, IRed, IMagenta, IBrown, IWhite, IDarkGrey, ILightBlue,
ILightCyan, ILightRed, ILightMagenta, ILightYellow, IBrWhite, IBlink AS INTEGER

   'M=Monitor%          'get the monitor type
   Answer% = 1
   CLS
```

```
CLOSE
SELECT CASE M
   CASE 3
      SCREEN 2
      VIEW (20, 20)-(590, 180)', IBlue, ILightBlue
      LINE (0, 0)-(540, 0)
      LINE (540, 0)-(540, 130)
      LINE (540, 130)-(0, 130)
      LINE (0, 130)-(0, 0)
      ScrnWidth% = 1000
      ScrnHeight% = 120
      OffSetX% = 0
      OffSetY% = ScrnHeight% \ 2
   CASE 4
      SCREEN 9
      VIEW (55, 55)-(595, 265), IBrWhite, IBrWhite
      VIEW (50, 50)-(590, 260), IBlue, ILightBlue
      LINE (0, 0)-(540, 0)
      LINE (540, 0)-(540, 210)
      LINE (540, 210)-(0, 210)
      LINE (0, 210)-(0, 0)
      ScrnWidth% = 1000
      ScrnHeight% = 180
      OffSetX% = 0
      OffSetY% = ScrnHeight% \ 2
   CASE 5
      SCREEN 9
      VIEW (55, 55)-(595, 265), IBrWhite, IBrWhite
      VIEW (50, 50)-(590, 260), IBlue, ILightBlue
      LINE (0, 0)-(540, 0)
      LINE (540, 0)-(540, 210)
      LINE (540, 210)-(0, 210)
      LINE (0, 210)-(0, 0)
      ScrnWidth% = 1000
      ScrnHeight% = 180
      OffSetX% = 0
      OffSetY% = ScrnHeight% \ 2
   END SELECT
WHILE Answer%

   OPEN "d:\segtec.dat" FOR OUTPUT AS #2

   IF INumSeg > 1 THEN
      OPEN "d:\~Geom.tmp" FOR INPUT AS #1
      INPUT #1, MaxX, MaxY
      MinX = MaxX
      MinY = MaxY
      FOR i = 1 TO INumSeg - 1
         INPUT #1, TmpX, TmpY
         IF ABS(TmpX) >= ABS(MaxX) THEN MaxX = TmpX
         IF ABS(TmpX) < ABS(MinX) THEN MinX = TmpX
```

```
            IF ABS(TmpY) >= ABS(MaxY) THEN MaxY = TmpY
            IF ABS(TmpY) < ABS(MinY) THEN MinY = TmpY
         NEXT i
         ScaleX = (ScrnWidth% / (MaxX - MinX))
         ScaleY = (ScrnHeight% / (MaxY - MinY)) / 2
         CLOSE #1
         OPEN "d:\~Geom.tmp" FOR INPUT AS #1
         FOR i = 1 TO INumSeg
            INPUT #1, TmpX, TmpY
            CIRCLE (INT(TmpX * ScaleX) + OffSetX%, INT(TmpY * ScaleY) + OffSetY%), 1
            PRINT #2, USING "###.####^^^^"; CSNG(TmpX);
            PRINT #2, USING "  ###.####^^^^"; CSNG(TmpY)
         NEXT i
         LOCATE 22, 1
         PRINT "XMax = "; MaxX, "YMax = "; MaxY
         PRINT "XMin = "; MinX, "YMin = "; MinY
      CLOSE
      NoYes Answer%, "RePlot Geometry?", 2
      IF Answer% = 1 THEN CLS : Answer% = -999 ELSE OPEN "d:\~Geom.tmp" FOR OUTPUT
AS #5: CLOSE #5
   WEND

END SUB


'*****************************************************************************
'* Screen0 - First Screen detailing the title and date of revision
'*****************************************************************************

SUB Screen0 (INumSections)

SHARED IBlack, IBlue, IGreen, ICyan, IRed, IMagenta, IBrown, IWhite, IDarkGrey, ILightBlue,
ILightCyan, ILightRed, ILightMagenta, ILightYellow, IBrWhite, IBlink AS INTEGER

   COLOR 15, 1: CLS
   'M = Monitor%            'get the monitor type
   IF M = 3 THEN COLOR IBrWhite, IBlue, IBlue ELSE COLOR IBrWhite, IBlue
   'COLOR IBrWhite, IBlue
   Centre "University of Alberta", 3
   COLOR 11, 1: Centre "[illegible] SEGMENTAL TECHNIQUE [illegible]", 6: COLOR 15, 1
   Centre "[illegible] Victor Tam [illegible]", 8
   Centre "August, 1990  Version 2.0", 10
   Centre "^P Strike a Key to Begin ^Q", 22
   Message$ = " SEGMENTAL  SHOOTING  TECHNIQUE ... Segmental Shooting Technique ..."
   Scroll Message$, 14, 112
   CLS
   LOCATE 10, 15
   INPUT "Please Input Number of Sections on Structurd: [1] ", INumSections
   IF INumSections = 0 THEN INumSections = 1

END SUB
```

```
'*****************************************************************************
'* Screen1        -        Second Screen inputing material properties, length, radius of curvature, modulus
                           of
'*                         rigidity, numerical tolerance, and number of segments
'*****************************************************************************
'
'
SUB Screen1 (Length(), RadCur(), INumSeg(), E(), DInertia(), EI(), TOL(), INumSections, ISecNum)

SHARED IBlack, IBlue, IGreen, ICyan, IRed, IMagenta, IBrown, IWhite, IDarkGrey, ILightBlue,
ILightCyan, ILightRed, ILightMagenta, ILightYellow, IBrWhite, IBlink AS INTEGER

   CLS
   Centre "University of Alberta", 3
   COLOR 11, 1: Centre "[redacted] SEGMENTAL TECHNIQUE [redacted]", 5: COLOR 15, 1
   Centre "[redacted] Victor Tam June, 1990 [redacted]", 7
   COLOR 11, 1: Centre "Section", 9
   PRINT ISecNum: COLOR 15, 1
   Centre "Please Input Structure Geometry & Material Properties", 11
   PRINT : PRINT

         '*** Set up Defaults for Section 1 ***

   IF ISecNum = 1 THEN
      INPUT "Please Input Length in metres: [1.0m] ", Length(ISecNum)
      IF Length(ISecNum) = 0 THEN Length(ISecNum) = 1
      INPUT "Please Input Young's Modulus in Pa [207000E+06]", E(ISecNum)
      IF E(ISecNum) = 0# THEN E(ISecNum) = 207000000000#
      INPUT "Please Input Moment of Inertia in m^4 [5.208E-11]", DInertia(ISecNum)
      IF DInertia(ISecNum) = 0# THEN DInertia(ISecNum) = (.005 ^ 4) / 12
      INPUT "Please Input Number of Segments [1000] ", INumSeg(ISecNum)
      IF INumSeg(ISecNum) = 0 THEN INumSeg(ISecNum) = 1000
      INPUT "Please Input Radius of Curvature [1E65] ", RadCur(ISecNum)
      IF RadCur(ISecNum) = 0 THEN RadCur(ISecNum) = 1D+65
      INPUT "Please Input Iteration Tolerance [1E-12] ", TOL(ISecNum)
      IF TOL(ISecNum) = 0 THEN TOL(ISecNum) = .000000000001#
      EI(ISecNum) = E(ISecNum) * DInertia(ISecNum)
   ELSE
      PRINT "Please Input Length in metres: "; "["; RIGHT$(STR$(Length(1)), LEN(STR$(Length(1)))
- 1); "]";
      INPUT Length(ISecNum)
      IF Length(ISecNum) = 0 THEN Length(ISecNum) = Length(1)
      PRINT "Please Input Young's Modulus in Pa ": "["; RIGHT$(STR$(E(1)), LEN(STR$(E(1))) - 1);
"]";
      INPUT E(ISecNum)
      IF E(ISecNum) = 0# THEN E(ISecNum) = E(1)
      PRINT "Please Input Moment of Inertia in m^4 ": "["; RIGHT$(STR$(DInertia(1)),
LEN(STR$(DInertia(1))) - 1); "]";
      INPUT DInertia(ISecNum)
      IF DInertia(ISecNum) = 0# THEN DInertia(ISecNum) = DInertia(1)
```

```
      PRINT "Please Input Number of Segments "; "["; RIGHT$(STR$(INumSeg(1)),
LEN(STR$(INumSeg(1))) - 1); "]";
      INPUT INumSeg(ISecNum)
      IF INumSeg(ISecNum) = 0 THEN INumSeg(ISecNum) = INumSeg(1)
      PRINT "Please Input Radius of Curvature "; "["; RIGHT$(STR$(RadCur(1)),
LEN(STR$(RadCur(1))) - 1); "]";
      INPUT RadCur(ISecNum)
      IF RadCur(ISecNum) = 0 THEN RadCur(ISecNum) = RadCur(1)
      PRINT "Please Input Iteration Tolerance "; "["; RIGHT$(STR$(TOL(1)), LEN(STR$(TOL(1))) -
1); "]";
      INPUT TOL(ISecNum)
      IF TOL(ISecNum) = 0 THEN TOL(ISecNum) = TOL(1)
      EI(ISecNum) = E(ISecNum) * DInertia(ISecNum)
   END IF

END SUB


'*****************************************************************************************
'* Screen2 - Third Screen prompting input pertaining to the loading conditions.
'*
'*          1 - Uniform Distributed Load
'*          2 - Uniform Normal Load
'*          3 - Concentrated Load
'*
'*****************************************************************************************
'
'
SUB Screen2 (q(), ILoadCond, ISecNum)

DIM MenuLoad$(3), IFlagLoad(3)
SHARED IBlack, IBlue, IGreen, ICyan, IRed, IMagenta, IBrown, IWhite, IDarkGrey, ILightBlue,
ILightCyan, ILightRed, ILightMagenta, ILightYellow, IBrWhite, IBlink AS INTEGER

   CLS
   IFlag1 = 1
   Entry$ = ""
   StartRow = 10

   'ILoadCond = Loading Conditions: 1 = Uniform Distributed Load
   '                                2 = Uniform Normal Load
   '                                3 = No Uniform Load

   Centre "Select Item(s) and Please Input Loading Conditions", 6

   MenuLoad$(1) = "1)   Non-Dimension Uniform Distributed Load q"
   MenuLoad$(2) = "2)   Non-Dimension Uniform Normal Load q"
   MenuLoad$(3) = "3)   No Distributed Load"

   'StartNode(1) = X1 value
   'StartNode(2) = Y1 value
```

```
'StartNode(3) = Gamma value
'StartNode(4) = M1 value
'StartNode(5) = V1 value
'StartNode(6) = T1 value

FOR i = 1 TO 3
   LOCATE StartRow + 4 * (i - 1), 20
   PRINT MenuLoad$(i)
NEXT i
PRINT
PRINT "Select Item:";

WHILE (IFlag1)

   WHILE (Entry$ = "")
      Entry$ = INKEY$
   WEND

   SELECT CASE Entry$
      CASE "1"
         IF IFlagLoad(1) = 0 THEN
            PRINT
            INPUT "Uniform Distributed Load (Heavy Elastica)"; q(ISecNum)
            IUnKnown = IUnKnown + 1
            IFlagLoad(1) = 1
            COLOR IGreen
            LOCATE StartRow, 20
            PRINT MenuLoad$(1)
            COLOR IBrWhite
            IFlag1 = 0
            ILoadCond = 1
         END IF
      CASE "2"
         IF IFlagLoad(2) = 0 THEN
            PRINT
            INPUT "Uniform Normal Load"; q(ISecNum)
            q(ISecNum) = SQR(q(ISecNum))
            IUnKnown = IUnKnown + 1
            IFlagLoad(2) = 1
            COLOR IGreen
            LOCATE StartRow + 2 * (1), 20
            PRINT MenuLoad$(2)
            COLOR IBrWhite
            IFlag1 = 0
            ILoadCond = 2
         END IF
      CASE "3"
         IF IFlagLoad(3) = 0 THEN
            PRINT
            IFlagLoad(3) = 1
            COLOR IGreen
```

```
            LOCATE StartRow + 2 * (1), 20
            PRINT MenuLoad$(2)
            COLOR IBrWhite
            IFlag1 = 0
            ILoadCond = 3
            q = 0
          END IF
        CASE ELSE
          SOUND 6000, 1
          PRINT
          Centre "Invalid Entry, Please Re-enter", 20
          IFlag1 = 1
      END SELECT

      Entry$ = ""
      LOCATE StartRow + 4 * (i - 1) - 2, 1
      PRINT "Select Item:";
      LOCATE StartRow + 4 * (i - 1) + 1, 1
      PRINT "
      LOCATE StartRow + 4 * (i - 1) - 2, 1
      PRINT "Select Item:";

      WEND

END SUB

'*****************************************************************************
'* Screen3 - Fourth Screen showing the input data for confirmation
'*****************************************************************************

SUB Screen3 (Length(), RadCur(), INumSeg(), E(), DInertia(), TOL(), StartNode(), StartNode$(),
IFlagStartNode(), ISecNum, INumSections)

SHARED TimeIdx AS STRING

   CLS : Length = 0
   OPEN "d:\segtec.out" FOR OUTPUT AS #3
   LOCATE 2, 1
      FOR i = 1 TO INumSections STEP 1
      COLOR 11, 1: PRINT "Section"; i: COLOR 15, 1
      PRINT
      PRINT "Length = ";
      PRINT USING "##.##"; Length(i);
      PRINT " m"
      PRINT "Young's Modulus = "; E(i) / 1000000#; " MPa"
      PRINT "Moment of Inertia = ";
      PRINT USING "  ##.##"; DInertia(i) * 1000 ^ 4;
      PRINT " mm^4"
      PRINT "Total Number of Segments = "; INumSeg(i)
```

```
    PRINT "Radius of Curvature = ";
    PRINT USING "  ##.##": RadCur(i);
    PRINT " m"
    PRINT "Tolerance = ";
    PRINT USING "+#.##^^^^": TOL(i)
    PRINT
    IF i < > 1 THEN
       IF (i / 2 - i \ 2) = 0 THEN
          Centre "Strike Any Key", 20
          SLEEP
          CLS
       END IF
    END IF
    NEXT i
    IF (i - 1 / 2) - (i - 1) \ 2 > 0 THEN
       Centre "Strike Any Key To Initiate Execution", 20
       SLEEP
       CLS
    END IF

    LOCATE 1, 1
    PRINT "Datd: "; DATE$
    TimeIdx$ = TIME$
    PRINT "Time Start: "; TimeIdx$
    LOCATE 8, 1
    PRINT "Known Parameter(s)", , "UnKnown Parameter(s)"
FOR i = 1 TO INumSections STEP 1
    PRINT #3, "Datd: "; DATE$
    PRINT #3, "Time Started: "; TimeIdx$
    PRINT #3, "Section"; i
    PRINT #3,
    PRINT #3, "Length = ";
    PRINT #3, USING "##.##"; Length(i);
    PRINT #3, " m"
    PRINT #3, "Young's Modulus = "; E(i) / 1000000#; " MPa"
    PRINT #3, "Moment of Inertia = ";
    PRINT #3, USING "  ##.##"; DInertia(i) * 1000 ^ 4;
    PRINT #3, " mm^4"
    PRINT #3, "Number of Segments = "; INumSeg(i)
    PRINT #3, "Radius of Curvature = ";
    PRINT #3, USING "  ##.##"; RadCur(i);
    PRINT #3, " m"
    PRINT #3, "Tolerance = ";
    PRINT #3, USING "+#.##^^^^"; TOL(i)
    PRINT #3,
    PRINT #3,
    PRINT #3,
NEXT i

j1 = 1
j2 = 1
```

```
    FOR i = 1 TO 6
      SELECT CASE IFlagStartNode(i)
        CASE 1
          LOCATE 10 + j1, 1
          PRINT StartNode$(i); " = ";
          PRINT USING "+#.############^^^^"; StartNode(i)
          'PRINT #3, StartNode$(i); " = ";
          'PRINT #3, USING "+#.############^^^^"; StartNode(i)
          j1 = j1 + 1
        CASE 2
          LOCATE 10 + j2, 44
          PRINT StartNode$(i); " = ";
          PRINT USING "+#.############^^^^"; StartNode(i)
          'PRINT #3, StartNode$(i); " = ";
          ' PRINT #3, USING "+#.############^^^^"; StartNode(i)
          j2 = j2 + 1
      END SELECT
    NEXT i

END SUB

SUB Scroll (Message$, IRow, Icolor)

    DO
      X$ = INKEY$
      Marquee Message$, IRow, Icolor    'do it on row 24 in black on white 112
    LOOP UNTIL LEN(X$)

END SUB

'*****************************************************************************
'* SelectParEnd - Assign the known boundary conditon(s) to the corresponding array(s).
'*****************************************************************************

SUB SelectParEnd (i, SecondKnown1, SecondKnown2, SecondKnown3, IShootDim, EndNode(), IFlag3())

    SELECT CASE IShootDim
      CASE 1
        SecondKnown1 = EndNode(i)
        IFlag3(1) = 1
      CASE 2
        IF IFlag3(2) = 0 THEN
          SecondKnown1 = EndNode(i)
          IFlag3(2) = IFlag3(2) + 1
        ELSE
          SecondKnown2 = EndNode(i)
          IFlag3(2) = IFlag3(2) + 1
        END IF
      CASE 3
        SELECT CASE IFlag3(3)
```

```
            CASE 0
               SecondKnown1 = EndNode(i)
               IFlag3(3) = IFlag3(3) + 1
            CASE 1
               SecondKnown2 = EndNode(i)
               IFlag3(3) = IFlag3(3) + 1
            CASE 2
               SecondKnown3 = EndNode(i)
               IFlag3(3) = IFlag3(3) + 1
            CASE ELSE
               PRINT "Something is Wrong"
            END SELECT
      END SELECT

END SUB

'*****************************************************************************
'* SelectParStart - Assign the known boundary conditon(s) to the corresponding array(s).
'*****************************************************************************

SUB SelectParStart (i, Z1(), Z2(), Z3(), IShootDim, StartNode(), IFlag2())

CONST SmallDiff = .005#

   SELECT CASE IShootDim
      CASE 1
         Z1(0) = StartNode(i)
         Z1(1) = StartNode(i) + StartNode(i) * SmallDiff
         IFlag2(1) = 1
      CASE 2
         IF IFlag2(2) = 0 THEN
            Z2(0, 1) = StartNode(i)
            Z2(1, 1) = StartNode(i) + StartNode(i) * SmallDiff
            IFlag2(2) = IFlag2(2) + 1
         ELSE
            Z2(0, 2) = StartNode(i)
            Z2(1, 2) = StartNode(i) + StartNode(i) * SmallDiff
            IFlag2(2) = IFlag2(2) + 1
         END IF
      CASE 3
         SELECT CASE IFlag2(3)
            CASE 0
               Z3(0, 1) = StartNode(i)
               Z3(1, 1) = StartNode(i) + StartNode(i) * SmallDiff
               IFlag2(3) = IFlag2(3) + 1
            CASE 1
               Z3(0, 2) = StartNode(i)
               Z3(1, 2) = StartNode(i) + StartNode(i) * SmallDiff
               IFlag2(3) = IFlag2(3) + 1
```

```
            CASE 2
               Z3(0, 3) = StartNode(i)
               Z3(1, 3) = StartNode(i) + StartNode(i) * SmallDiff
               IFlag2(3) = IFlag2(3) + 1
            CASE ELSE
               PRINT "Something is Wrong"
            END SELECT
         END SELECT

END SUB

'*****************************************************************************
'* SolveForceDistLd - The segmental approach for solving rod problems with Distributed Loading.
'*****************************************************************************

SUB SolveForceDistLd (Parameters(), Delta, EI, RadCur, INumSeg, TOL, Energy, q)

DIM Alpha(1000)

   'StartNode(1) = X1 value;      Parameters(1)
   'StartNode(2) = Y1 value;      Parameters(2)
   'StartNode(3) = Gamma value;   Parameters(3)
   'StartNode(4) = M1 value;      Parameters(4)
   'StartNode(5) = V1 value;      Parameters(5)
   'StartNode(6) = T1 value;      Parameters(6)

   OPEN "d:\curve.dat" FOR OUTPUT AS #4

   'q = 8.5
   ww = q ^ 2 * EI / (Delta * INumSeg) ^ 3'-1                    'Weight/Unit
Length
   W = ww * (Delta * INumSeg) ^ 3 / EI                  'Non-Dimensional Weight/Unit Length (N.B.
Delta*INumSeg = Length)
   'Parameters(5) = Parameters(5) + -ww * Delta * INumSeg   'Assign V1 to account for Distributed
Load
   Energy = 0#                                  'Initilaize Strain Energy Variable
   j = 0                                        'Initilaize Segment# Looping Index

   'IF INumSeg > 1 THEN OPEN "d:\~Geom.Tmp" FOR OUTPUT AS #1

   WHILE (j < INumSeg)

      j = j + 1

'***    Non-Dimensionalize Parameters

      Mu1 = Parameters(4) * Delta / EI
      Nu1 = Parameters(5) * (Delta ^ 2) / EI
      Tau1 = Parameters(6) * (Delta ^ 2) / EI
```

```
X = Parameters(1) / Delta
Y = Parameters(2) / Delta
MuE = Mu1 + Delta / RadCur
Beta = -(3 * MuE + Nu1) / 6
Wx = -ww * SIN(Parameters(3) - Beta)
Wy = ww * COS(Parameters(3) - Beta)
DKai = Wx * (Delta ^ 3) / EI
Sai = Wy * (Delta ^ 3) / EI

'***  Assign Power Series Terms Alpha0 - Alpha3

k = 4
Alpha(k - 4) = Beta                                    'Alpha0
Alpha(k - 3) = Mu1 + Delta / RadCur                    'Alpha1
Alpha(k - 2) = Nu1 / 2                                 'Alpha2
Alpha(k - 1) = (Tau1 * (Mu1 + Delta / RadCur) - DKai * Beta - Sai) / 6     'Alpha3
Phi = Alpha(k - 4) + Alpha(k - 3) + Alpha(k - 2) + Alpha(k - 1)
Mu2 = Alpha(k - 3) + 2 * Alpha(k - 2) + 3 * Alpha(k - 1)
Nu2 = 3 * 2 * Alpha(k - 1) + 2 * Alpha(k - 2)
DeltaH = Alpha(k - 4) + Alpha(k - 3) / 2 + Alpha(k - 2) / 3 + Alpha(k - 1) / 4

DO

   Alpha(k) = (Tau1 * Alpha(k - 2) - DKai * Alpha(k - 3)) / (k * (k - 1))
   Phi = Phi + Alpha(k)
   Mu2 = Mu2 + k * Alpha(k)
   DeltaH = DeltaH + Alpha(k) / (k + 1)
   Nu2 = Nu2 + k * (k - 1) * Alpha(k)
   Alpha(k - 3) = Alpha(k - 2)
   Alpha(k - 2) = Alpha(k - 1)
   Alpha(k - 1) = Alpha(k)
   k = k + 1

LOOP UNTIL (ABS(Alpha(k - 1) + Alpha(k - 2) + Alpha(k)) < TOL)

Mu2 = Mu2 - Delta / RadCur
X = X + (COS(Parameters(3) - Beta) - DeltaH * SIN(Parameters(3) - Beta))
Y = Y + (SIN(Parameters(3) - Beta) + DeltaH * COS(Parameters(3) - Beta))
Rho2 = X' / Delta
Tau2 = Tau1 + Nu1 * Beta - Nu2 * Phi - DKai    'Rho2 = 1? (EndNode?)
Parameters(3) = Parameters(3) + Phi - Beta
M2 = Mu2 * EI / Delta
T2 = Tau2 * EI / (Delta ^ 2)
V2 = Nu2 * EI / (Delta ^ 2)
Parameters(1) = X * Delta
Parameters(2) = Y * Delta
Parameters(4) = M2
Parameters(6) = T2
Parameters(5) = V2
LOCATE 1, 54
IF (j / 50 - (j \ 50)) = 0 THEN
```

```
        SELECT CASE kk
          CASE 0
            PRINT "- SEGMENT# "; j; "    "
            kk = kk + 1
          CASE 1
            PRINT "\ SEGMENT# "; j; "    "
            kk = kk + 1
          CASE 2
            PRINT "| SEGMENT# "; j; "    "
            kk = kk + 1
          CASE 3
            PRINT "/ SEGMENT# "; j; "    "
            kk = 0
          CASE ELSE
            PRINT "Something is Wrong!"
        END SELECT
      END IF

      PRINT #4, j, CSNG(NonDCurvature), CSNG(DCurvature)

      PRINT #1, Parameters(1), Parameters(2)

    WEND

      LOCATE 17, 1

END SUB

'*****************************************************************************************
'* SolveForceNormLd - The segmental approach for solving rod problems with Normal Loading.
'*****************************************************************************************

SUB SolveForceNormLd (Parameters(), Delta, EI, RadCur, INumSeg, TOL, Energy, q)

DIM Alpha(1000)


    'StartNode(1) = X1 value:      Parameters(1)
    'StartNode(2) = Y1 value:      Parameters(2)
    'StartNode(3) = Gamma value:   Parameters(3)
    'StartNode(4) = M1 value:      Parameters(4)
    'StartNode(5) = V1 value:      Parameters(5)
    'StartNode(6) = T1 value:      Parameters(6)

    pp = q ^ 2 * EI / (Delta * INumSeg) ^ 3          'Weight/Unit Length
    Energy = 0#                                      'Initilaize Strain Energy Variable
    j = 0                                            'Initilaize Segment# Looping Index
```

```
WHILE (j < INumSeg)

    j = j + 1

'***    Non-Dimensionalize Parameters

    Mu1 = Parameters(4) * Delta / EI
    Nu1 = Parameters(5) * (Delta ^ 2) / EI
    Tau1 = Parameters(6) * (Delta ^ 2) / EI
    X = Parameters(1) / Delta
    Y = Parameters(2) / Delta
    MuE = Mu1 + Delta / RadCur
    Beta = -(3 * MuE + Nu1) / 6
    Wt = 0                                  'Tangential Component
    Wn = pp                                 'Normal Component
    Eta = Wn * (Delta ^ 3) / EI
    Zeta = Wt * (Delta ^ 3) / EI

'***    Assign Power Series Terms Alpha0 - Alpha3

    k = 4
    Alpha(k - 4) = Beta                     'Alpha0
    Alpha(k - 3) = Mu1 + Delta / RadCur     'Alpha1
    Alpha(k - 2) = Nu1 / 2                  'Alpha2
    Alpha(k - 1) = (Tau1 * (Mu1 + Delta / RadCur) - Eta) / 6     'Alpha3
    Phi = Alpha(k - 4) + Alpha(k - 3) + Alpha(k - 2) + Alpha(k - 1)
    Mu2 = Alpha(k - 3) + 2 * Alpha(k - 2) + 3 * Alpha(k - 1)
    Nu2 = 3 * 2 * Alpha(k - 1) + 2 * Alpha(k - 2)
    DeltaH = Alpha(k - 4) + Alpha(k - 3) / 2 + Alpha(k - 2) / 3 + Alpha(k - 1) / 4

    DO
        Alpha(k) = (Tau1 * Alpha(k - 2) - Zeta * (1 - 1 / (k - 2)) * Alpha(k - 3)) / (k * (k - 1))
        Phi = Phi + Alpha(k)
        Mu2 = Mu2 + k * Alpha(k)
        DeltaH = DeltaH + Alpha(k) / (k + 1)
        Nu2 = Nu2 + k * (k - 1) * Alpha(k)
        Alpha(k - 3) = Alpha(k - 2)
        Alpha(k - 2) = Alpha(k - 1)
        Alpha(k - 1) = Alpha(k)
        k = k + 1
    LOOP UNTIL (ABS(Alpha(k - 1) + Alpha(k - 2) + Alpha(k)) < TOL)

    Mu2 = Mu2 - Delta / RadCur
    X = X + (COS(Parameters(3) - Beta) - DeltaH * SIN(Parameters(3) - Beta))
    Y = Y + (SIN(Parameters(3) - Beta) + DeltaH * COS(Parameters(3) - Beta))
    Rho2 = X
    Tau2 = Tau1 + Nu1 * Beta - Nu2 * Phi - Zeta - Eta * Alpha(k - 1) / k'* Y / Delta     'Rho2 =
1? (EndNode?)
    Parameters(3) = Parameters(3) + Phi - Beta
    M2 = Mu2 * EI / Delta
    T2 = Tau2 * EI / (Delta ^ 2)
```

```
        V2 = Nu2 * EI / (Delta ^ 2)
        Parameters(1) = X * Delta
        Parameters(2) = Y * Delta
        Parameters(4) = M2
        Parameters(6) = T2
        Parameters(5) = V2
        LOCATE 1, 54
        IF (j / 50 - (j \ 50)) = 0 THEN
           SELECT CASE kk
              CASE 0
                 PRINT "- SEGMENT# "; j;"\"    "
                 kk = kk + 1
              CASE 1
                 PRINT "\ SEGMENT# "; j; "    "
                 kk = kk + 1
              CASE 2
                 PRINT "| SEGMENT# "; j; "    "
                 kk = kk + 1
              CASE 3
                 PRINT "/ SEGMENT# "; j; "    "
                 kk = 0
              CASE ELSE
                 PRINT "Something is Wrong!"
           END SELECT
        END IF

           PRINT #1, Parameters(1), Parameters(2)
     WEND
     LOCATE 17, 1

END SUB

'*****************************************************************************************
'* SolveForcePtLd - The segmental approach for solving rod problems with Concentrated Loading.
'*****************************************************************************************

SUB SolveForcePtLd (Parameters(), Delta, EI, RadCur, INumSeg, TOL, Energy, q)

DIM Alpha(500)

   'StartNode(1) = X1 value;      Parameters(1)
   'StartNode(2) = Y1 value;      Parameters(2)
   'StartNode(3) = Gamma value;   Parameters(3)
   'StartNode(4) = M1 value;      Parameters(4)
   'StartNode(5) = V1 value;      Parameters(5)
   'StartNode(6) = T1 value;      Parameters(6)

   Energy = 0#
   j = 0
```

```
WHILE (j < lNumSeg)

   j = j + 1
   Mu1 = Parameters(4) * Delta / EI
   Nu1 = Parameters(5) * (Delta ^ 2) / EI
   Tau1 = Parameters(6) * (Delta ^ 2) / EI
   X = Parameters(1) / Delta
   Y = Parameters(2) / Delta
   MuE = Mu1 + Delta / RadCur
   Beta = -(3# * MuE + Nu1) / 6#
   k = 3
   Alpha(k - 3) = Beta
   Alpha(k - 2) = Mu1 + Delta / RadCur
   Alpha(k - 1) = Nu1 / 2#
   Fay = Alpha(k - 3) + Alpha(k - 2) + Alpha(k - 1)
   Mu2 = Alpha(k - 2) + 2# * Alpha(k - 1)
   Nu2 = 2# * Alpha(k - 1)
   DeltaH = Alpha(k - 3) + Alpha(k - 2) / 2# + Alpha(k - 1) / 3#

   DO
      Alpha(k) = Tau1 * Alpha(k - 2) / (k * (k - 1))
      Fay = Fay + Alpha(k)
      Mu2 = Mu2 + k * Alpha(k)
      DeltaH = DeltaH + Alpha(k) / (k + 1)
      Nu2 = Nu2 + k * (k - 1) * Alpha(k)
      k = k + 1
   LOOP UNTIL (ABS(Alpha(k - 1) + Alpha(k - 2) + Alpha(k - 3)) < TOL)

   Mu2 = Mu2 - Delta / RadCur
   Tau2 = Tau1 + Nu1 * Beta - Nu2 * Fay
   X = X + (COS(Parameters(3) - Beta) - DeltaH * SIN(Parameters(3) - Beta))
   Y = Y + (SIN(Parameters(3) - Beta) + DeltaH * COS(Parameters(3) - Beta))
   Parameters(3) = Parameters(3) + Fay - Beta
   M2 = Mu2 * EI / Delta
   T2 = Tau2 * EI / (Delta ^ 2)
   V2 = Nu2 * EI / (Delta ^ 2)
   Parameters(1) = X * Delta
   Parameters(2) = Y * Delta
   Parameters(4) = M2
   Parameters(6) = T2
   Parameters(5) = V2
   LOCATE 1, 54
   IF (j / 50 - (j \ 50!)) = 0 THEN
      SELECT CASE kk
         CASE 0
            PRINT "- SEGMENT# "; j; "   "
            kk = kk + 1
         CASE 1
            PRINT "\ SEGMENT# "; j; "   "
            kk = kk + 1
         CASE 2
```

```
            PRINT "| SEGMENT# "; j; "    "
            kk = kk + 1
          CASE 3
            PRINT "/ SEGMENT# "; j; "    "
            kk = 0
          CASE ELSE
            PRINT "Something is Wrong!"
        END SELECT
      END IF
      WRITE #1, Parameters(1), Parameters(2)
    WEND
    LOCATE 17, 1

END SUB

'*****************************************************************************
'* SolveForces   -    Invoke the corresponding solveing subroutine base on the Loading Condition
'*                    (Point,
'*                    Distributed and Normal Load)
'*****************************************************************************

SUB SolveForces (ILoadCond, Parameters(), Delta, EI, RadCur, INumSeg, TOL, Energy, q)

SELECT CASE ILoadCond
   CASE 2
      SolveForceNormLd Parameters(), Delta, EI, RadCur, INumSeg, TOL, Energy, q
   CASE 1
      SolveForceDistLd Parameters(), Delta, EI, RadCur, INumSeg, TOL, Energy, q
   CASE 3
      SolveForcePtLd Parameters(), Delta, EI, RadCur, INumSeg, TOL, Energy, q
END SELECT

END SUB
```